(12)

AD-A148 451

OCTOBER 1984 LIDS-TH-1416

Research Supported By:

Defense Advanced Research Projects Agency
Contract N00014-84-K-0357

TRANSMISSION SCHEDULING TO REDUCE CONVEX DELAY COSTS IN PACKET NETWORKS

Darius M. Thabit

Laboratory for Information and Decision Systems
MASSACHUSETTS INSTITUTE OF TECHNOLOGY, CAMBRIDGE, MASSACHUSETTS 02139

October 1984 LIDS-TH-1416

TRANSMISSION SCHEDULING TO REDUCE CONVEX DELAY COSTS IN PACKET NETWORKS

by

Darius M. Thabit

This report is based on the unaltered thesis of Darius M. Thabit submitted in partial fulfillment of the requirements for the degree Master of Science at the Massachusetts Institute of Technology in February 1984. This research was conducted at the M.I.T. Laboratory for Information and Decision Systems with partial support provided by the Defense Advanced Research Projects Agency under Contract N00014-84-K-0357.

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A148 451	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) TRANSMISSION SCHEDULING TO REDUCE CONVEX DELAY COSTS IN PACKET NETWORKS		5. TYPE OF REPORT & PERIOD COVERED Thesis
		6. PERFORMING ORG. REPORT NUMBER LIDS-TH-1416
7. AUTHOR(s) Darius M. Thabit		8. CONTRACT OR GRANT NUMBER(s) DARPA Order No. 3045/2-2-84 Amendment #11 ONR/N00014-84-K-0357
9. PERFORMING ORGANIZATION NAME AND ADDRESS Massachusetts Institute of Technology Laboratory for Information and Decision Systems Cambridge, Massachusetts 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Code No. 5T10 ONR Identifying No. 049-383
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209		12. REPORT DATE October 1984
		13. NUMBER OF PAGES 61
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research Information Systems Program Code 437 Arlington, Virginia 22217		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Delay is a fundamental issue in packet communication networks. Previous work has focused on expected delay as a performance measure in queueing models, in the context of routing. It is clear that some types of traffic are more sensitiv to delay than others, and some simple priority structures have been proposed; but the fact that expected delay is not very meaningful for certain applications (e.g. packetized voice, which has a critical delay ceiling) has received little attention.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (Continued).

Both issues are addressed here by defining traffic classes, and assigning delay costs $c(b,t)$ which reflect the undesirability of a class-b packet spending time t in the network. We argue that convex, nondecreasing (CND) functions are suitable for a variety of classes; examples suggest a range from linear (e.g. file transfers) to step-like (e.g. packetized voice). An appropriate performance measure is the limiting expected average cost per packet. (As an equivalent characterization, we show that, for a given class, the expected average cost converges to the expected cost in equilibrium, under mild conditions.) Within a given class, strictly convex costs will be driven by packets whose routes have many hops, and/or packets which have long delays on some links; such packets may profitably be given priority as well.

We propose a distributed scheme in which a single scheduling policy is executed independently on every link. Link traffic is modeled by a stochastic sequence $\{y_n\}$, where $\langle a_n,b_n,h_n,r_n,\tau_n\rangle \equiv y_n : \Omega Y \equiv \mathbb{R}\times B\times[0,\infty)\times R\times(0,\infty)$ represents for packet n the epoch of arrival to the link ($a_n \leq a_{n+1}$), traffic class, delay on upstream links, route, and transmission time; $y_n(\omega)$ becomes known to the link scheduler at $a_n(\omega)$. Routing-scheduling interactions and communication overhead are mentioned, and desirable properties for estimates f of the delay on downstream links are discussed. We develop a dynamic programming formulation in which the scheduler incurs a decision cost $g(t,y) \equiv c(b,h+t-a+\tau+f)$ by selecting for transmission at epoch t a packet with characteristics $y \in Y$.

We focus on priority rules $\phi(t,y)$: at epoch t, a packet for which ϕ is maximum is selected for transmission. A rule ϕ^* is optimal (in the static sense) if it empties out any given queue at minimum cost; e.g. for linear costs $g(t,y) = m_y t$, we shown that the "μC rule" $\phi = m_y/\tau$ is optimal in this sense. A significant result here is that, in the case of constant transmission times, an optimal time-independent rule minimizes the limiting-average cost for every ω; thus for linear decision costs, the rule $\phi = m_y$ generates an optimal policy for a G/D/1 link. For delay costs of the form $c(b,t) = c_0(t+\Delta_b)$, where c_0 is CND, we show similarly that the rule $\phi = \Delta_b+h-a+f$ is optimal for a G/D/1 link if f is time-independent. Suboptimal rules are also given for certain costs in the case of variable service times.

TRANSMISSION SCHEDULING TO REDUCE CONVEX DELAY COSTS IN PACKET NETWORKS

by
Darius Mitchell Thabit

S.B.E.E. Massachusetts Institute of Technology
(1983)

Submitted in partial fulfillment of the
requirements for the degrees of

Master of Science in Electrical Engineering

and

Electrical Engineer

at the

Massachusetts Institute of Technology

February 1984

Signature of Author ______________________
Department of Electrical Engineering
and Computer Science
14 October 1983

Certified by ______________________
Prof. Robert G. Gallager
Thesis Supervisor

Accepted by ______________________
Prof. Arthur C. Smith
Chairman, Committee on Graduate Students

TRANSMISSION SCHEDULING TO REDUCE CONVEX DELAY COSTS IN PACKET NETWORKS

by Darius M. Thabit

Submitted to the Massachusetts Institute of Technology on 14 October 1983, in partial fulfillment of the requirements for the degrees of Electrical Engineer, and Master of Science in Electrical Engineering

ABSTRACT

Delay is a fundamental issue in packet communication networks. Previous work has focused on expected delay as a performance measure in queueing models, in the context of routing. It is clear that some types of traffic are more sensitive to delay than others, and some simple priority structures have been proposed; but the fact that expected delay is not very meaningful for certain applications (*e.g.* packetized voice, which has a critical delay ceiling) has received little attention.

Both issues are addressed here by defining *traffic classes,* and assigning *delay costs* $c(b,t)$ which reflect the undesirability of a class-b packet spending time t in the network. We argue that convex, nondecreasing (CND) functions are suitable for a variety of classes; examples suggest a range from linear (*e.g.* file transfers) to step-like (*e.g.* packetized voice). An appropriate performance measure is the limiting expected average cost per packet. (As an equivalent characterization, we show that, for a given class, the expected average cost converges to the expected cost in equilibrium, under mild conditions.) Within a given class, strictly convex costs will be driven by packets whose routes have many hops, and/or packets which have long delays on some links; such packets may profitably be given priority as well.

We propose a distributed scheme in which a single scheduling policy is executed independently on every link. Link traffic is modeled by a stochastic sequence $\{y_n\}$, where $\langle a_n, b_n, h_n, r_n, \tau_n \rangle \equiv y_n : \Omega \to Y \equiv \mathbb{R} \times B \times [0,\infty) \times R \times (0,\infty)$ represents for packet n the epoch of arrival to the link ($a_n \leq a_{n+1}$), traffic class, delay on upstream links, route, and transmission time; $y_n(\omega)$ becomes known to the link scheduler at $a_n(\omega)$. Routing-scheduling interactions and communication overhead are mentioned, and desirable properties for estimates f of the delay on downstream links are discussed. We develop a dynamic programming formulation in which the scheduler incurs a *decision cost* $g(t,y) \equiv c(b, h+t-a+\tau+f)$ by selecting for transmission at epoch t a packet with characteristics $y \in Y$.

We focus on priority rules $\phi(t,y)$: at epoch t, a packet for which ϕ is maximum is selected for transmission. A rule ϕ^* is optimal (in the static sense) if it empties out any given queue at minimum cost; *e.g.* for linear costs $g(t,y) = m_y t$, we show that the "μC rule" $\phi = m_y/\tau$ is optimal in this sense. A significant result here is that, in the case of constant transmission times, an optimal time-independent rule minimizes the limiting-average cost for every ω; thus for linear decision costs, the rule $\phi = m_y$ generates an optimal policy for a G/D/1 link. For delay costs of the form $c(b,t) = c_0(t+\Delta_b)$, where c_0 is CND, we show similarly that the rule $\phi = \Delta_b + h - a + f$ is optimal for a G/D/1 link if f is time-independent. Suboptimal rules are also given for certain costs in the case of variable service times.

Thesis Supervisor: Robert G. Gallager
Title: Professor of Electrical Engineering

ACKNOWLEDGEMENTS

Robert Gallager's supervision, in particular the careful review and pointed criticism, has had a substantial impact on the quality of this thesis; I have been fortunate to have had the benefit of his experience and insight.

Life at MIT has often been trying, but a number of people have helped make it a home away from home, and so provided essential support for this effort. Michael Komichak and Rene Cruz have been especially close friends over the years, and inspiring classmates; Rene and Saul Gelfand dutifully allowed themselves to be drawn into thesis-related discussions. I salute LIN, JDA, THL, WLL, and MDC, fellow defenders of the video galaxy, and ground crew AM, EDT, and JS, all good friends as well. On behalf of every EECS graduate student, I thank Marilyn Pierce for her dedication and unlimited patience.

This work was financially supported in part by the MIT Laboratory for Information and Decision Systems, with a grant from the Defence Advanced Research Projects Agency.

TABLE OF CONTENTS

Abstract ii

Acknowledgements iii

List of Figures v

Chapter I - Introduction 1
- 1.1 Communication Delays in Packet Networks 1
- 1.2 Delay Performance Criteria 3

Chapter II - A Cost Structure for Packet Delays 6
- 2.1 Delay Cost Functions 6
- 2.2 Stochastic Performance Measures 11
- 2.3 Delay Performance in Equilibrium 14
  - 2.3.1 Packet Network Model 15
  - 2.3.2 Implications of Convex Delay Costs 17

Chapter III - Scheduling Algorithms for Packet Networks 22
- 3.1 A Decentralized Scheduling Problem 22
- 3.2 A Dynamic Programming Formulation of Link Scheduling 25
  - 3.2.1 Process State and Dynamics 26
  - 3.2.2 Decision Costs 28
  - 3.2.3 A Refinement of the Downstream Delay Estimate 30
  - 3.2.4 Performance on a Saturated Link 33
- 3.3 A Class of Efficient Scheduling Policies 36
  - 3.3.1 Myopic Scheduling Policies 36
  - 3.3.2 Priority Rules 38
- 3.4 Selection Rules for Strictly Convex Costs 42

Chapter IV - Summary and Conclusion 47

Appendix 49
- A.1 Convergence of Average Cost to Equilibrium Cost 49
- A.2 Alternate Route Delays in an Optimal Flow Assignment 51
- A.3 Optimal Priority-based Policies for G/D/1 Links 53

References 59

LIST OF FIGURES

Figure 1 Possible delay cost function for packetized voice 9
Figure 2 Possible delay cost function for display terminal 9
Figure 3 Density of system time in a hypothetical M/M/1 network 18
Figure 4 Elapsed times with pessimistic estimates 32
Figure 5 How myopic policies can fail 37
Figure 6 Shifted versions of a given nominal cost 45

CHAPTER I - INTRODUCTION

1.1 Communication Delays in Packet Networks

In many types of communication systems, the traffic load is not a static quantity; even during periods of peak average loading, the instantaneous load typically has a substantial variation about this average. However, providing a system with sufficient capacity to accommodate these instantaneous load peaks can almost never be justified economically, and the reduced allocation of system capacity may result in degradation of one or more aspects of service to the users.

In packet-switched systems, such as datagram or virtual-circuit communication networks, point-to-point transmission of data packets is accomplished over a single shared channel, and packets ready to depart are queued in a buffer when the transmitter on a link is busy [Tanenbaum; §3.3.5] [Heart *et al*]. Due to the irregular pattern in which data traffic arrives from outside the network, a certain degree of queueing due to entering traffic will be almost inevitable, and this will be compounded by internal traffic. In this case, then, the limited system capacity manifests itself in the form of queueing delays. Of course there are other delays associated with packet switching: however, packet processing and transmission delays are relatively small, and propagation delays become significant only in satellite applications; in any case, all of these are more-or-less constant system parameters as far as a given user-pair is concerned.

Communication delay is a fundamental performance issue in packet networks. As one consideration, many applications have certain necessary or desirable response-time characteristics for the individual packets. As an example, the perceived quality of a display terminal session is degraded as the

round-trip delay is increased; in particular, it is tempting to speculate that this kind of consideration will be relevant whenever the psychophysical needs of the user come into play. As another example, in certain kinds of real-time, distributed processing environments, the value of information decreases as it gets older.

As a second consideration, it is often the case that each of the individual packet delays contributes to the total duration of a given session or, more specifically, of a given job. In this context, the effects of delay are most noticeable in applications characterized by frequent exchanges of short messages; as the frequency of such exchanges increases, the delay becomes a larger fraction of the total time required to complete the job. At the other extreme, in a "one-sided" data conversation such as a file transfer, the sender may have many packets in the network simultaneously, in which case the effect of individual packet delays on total job time will be much weaker.[1]

It appears that, for many applications, delay considerations can be resolved into one or both of the basic types described above; and moreover, that one of these will often dominate the other. For instance, in a voice conversation, user-related constraints on the round-trip response times of individual packets should come into play decisively at relatively low levels of delay; again, one might suspect that this would apply in those cases in which a human is involved in a direct way [*cf.* Roberts *et al*; p. 544]. On the other hand, in certain types of machine-machine interactions, individual packet delays would be irrelevant, and the total job time will be the dominant consideration, as Roberts *et al*. [p. 544] have observed.

---

[1]As an intermediate case, in a "window" flow-control strategy, the sender is allowed no more than M packets in the network at one time [see Gerla *et al*, 1980]; the job time will then be determined essentially by the round-trip delays of every Mth packet.

The primary motivation for distinguishing these two types of applications is that they typically lead to different kinds of performance criteria, as discussed in §2.1. For brevity, applications for which the individual packet delays are the dominant consideration will be referred to here as *packet-sensitive,* and those for which the job time dominates will be called *job-sensitive.*

As a final consideration, from the network operator's point of view, congestion will result if packets spend too much time in the network. Although one could attempt to incorporate this into delay costs, we expect that buffering capability will be such that congestion is typically dominated by other considerations.

1.2 Delay Performance Criteria

Published work on packet networks has been concerned almost exclusively with reducing the average delay per packet. In particular, considerable progress has been made in the development of packet routing algorithms under the average delay criterion, due in part to the linearity of this performance measure in the individual link delays [see *e.g.* Gallager].

However, average delay as a performance measure has two serious weaknesses. First of all, desirable delay characteristics for data traffic will vary over a wide range for different applications. Roberts *et al.* [p. 544] give some figures for typical time-sharing applications as follows: a 50 character line of text (400 bits) should traverse the network in at most 200ms; for interactive graphics, a new display page (20 kbits) should take less than a second, and interrupts (less than 100 bits) within 30-90ms. According to Bell Labs [p. 56], voice conversation is not "unduly" impaired if the origin-destination delay is less than 300ms. Low priority items, such as certain kinds of file transfers or sensor data, could reasonably be expected to tolerate delays on the order

of several seconds or more. One recent analysis which addresses this issue provides for scheduling voice packets ahead of data packets [Ibe].

At a more fundamental level, however, for certain types of applications average delay simply is not very meaningful. This is perhaps best illustrated by an example such as the following: in a packetized voice transmission system, voice samples or "frames" must be delivered to the destination at regular intervals, to be reassembled and decoded for the listener; if a packet arrives early it is simply buffered; if it arrives late it must be discarded. It is fairly clear that the fraction of packets arriving "on time" is all we really care about here; depending on the distribution of packet delays, the average delay need not be a very reliable indicator of this.

Thus the figures cited in the next to last paragraph leave something to be desired; for instance, must we hold the delay of every text packet below 200ms, or will we be content to achieve this as an average value? More specifically, what are the consequences of exceeding this value, and what, if anything, do we gain by undercutting it? Such questions have received little if any attention in the literature.

For certain aspects of telephone system performance, Bell System specifications are reported to be established as follows. The expected percentage of users that would bestow a particular subjective rating (e.g. "good"), in conjunction with a particular aspect of raw performance (e.g. receiver rms noise voltage), is called the *grade of service* associated with that rating and performance aspect [Bell; pp. 45-48]; the calculation of grades of service is based in part on subjective testing of customers. Typical objectives are to provide grades of service of about 95% in the "good or better" category and a negligible fraction in the "poor" category.

The grade of service concept represents a step in the right direction because it reflects, in essence, a functional dependence of user satisfaction on

raw performance. Here, however, we are primarily interested in the automatic control of delay performance in an operational network, and this will require an objective function, *i.e.* a scalar measure of global performance, rather than a collection of individual specifications. It is common in an operations research context to assign costs as functions of delay for various commodities, with the expected cost as a performance measure. We adopt this approach here, and thus exploit the notion of user dependence in a very direct way.

We are motivated here by the philosophy that, for a given application, any user should be able to expect a certain level of performance from the system, regardless of geographical separation or other non-essential factors. As Wong *et al*. have pointed out, this is particularly reasonable when users are charged on the basis of resource usage. Our goal then is to provide acceptable service, for all users at as small a dollar cost as possible to the operator, or equivalently for as many users as possible at a given dollar expenditure.

In Chapter II, we develop a cost structure for packet delays, and show how it addresses the issues raised here. A probabilistic framework is defined, and by considering equilibrium behavior some qualitative insights are obtained. In Chapter III, we describe a distributed implementation of the basic scheduling idea, and construct a model for the sequential decision process associated with packet selection. We can show that in certain idealized cases, optimal policies are given by simple selection rules.

CHAPTER II - A COST STRUCTURE FOR PACKET DELAYS

2.1 Delay Cost Functions

Let B be some set of *data traffic classes:* for instance, B might be a set of applications, *e.g.* video terminal, packetized voice, *etc;* but B is not restricted to any particular interpretation, and we require only that it be a countable set. We assume the availability of a function c from $B\times[0,\infty)$ into the real numbers $\mathbb{R}$, such that c(b,t) represents the undesirability, from the standpoint of the entity operating the network, of a "typical" user experiencing an individual packet delay t for a class-b packet, $t\geq 0$, $b\in B$. For short, the function c(b,) is referred to simply as the *delay cost* for class-b packets. It isn't obvious how such functions would be constructed in practice. They should of course draw on subjective testing for applications directly involving humans; it would probably be most straightforward in such cases to construct for $t\geq 0$ the associated *marginal cost* function[2]

$$m(b,t) \equiv \partial c(b,t)/\partial t, \tag{2.1}$$

i.e. $m(b,t)\Delta t$ reflects the aproximate cost of increasing the delay of a class-b packet from t to $t+\Delta t$. Then the delay cost clearly may be derived as

$$c(b,t) = \int_0^t m(b,x)dx + c(b,0) \tag{2.2}$$

for some c(b,0).

Now consider a collection of packets, labelled (uniquely) by either the positive integers $\mathbb{N}^+$ or a subset $\{1,2,\ldots N\}$, which traverse some system;

[2]lacking the standard "equals by definition" symbol in our character set, we will make do with "≡".

suppose that packet n belongs to class $b_n \epsilon B$ and spends time $s_n \geq 0$ in the system. Obvious measures of performance are the total cost

$$\sum_{n=1}^{N} c(b_n, s_n) \tag{2.3}$$

in the finite case, and the long-term average cost per packet

$$\lim_{N \to \infty} (1/N) \sum_{n=1}^{N} c(b_n, s_n) \tag{2.4}$$

in the infinite case, assuming convergence. The additive structure without any weighting implicitly requires that the delay costs c(b,) reflect, as much as possible, the relative urgency of the various traffic classes. It also implies that time-independent constants in the delay costs are irrelevant to the minimization problem, so that we may as well take c(b,0) = 0 in (2.2).

In an attempt to get some feeling for the general behavior of delay cost functions, we will speculate on their form for some representative applications. At the same time, we will argue that functions which are convex and nondecreasing (or CND for short) are suitable as delay costs for a reasonably wide variety of applications; it turns out that CND cost functions have a number of desirable properties, and we want to exploit these in the sequel. We note here that for a convex cost c(b,), the associated marginal cost m(b,) is nondecreasing, and this answers in a general way the questions raised in §1.2 concerning constraints on raw delay: we lose at least as much by exceeding a given constraint as we would gain by undercutting it by the same amount [*cf.* Haji *et al*].

Consider first, as a packet-sensitive application, the packetized voice connection mentioned earlier in §1.2. The cost associated with the transit time of a single voice packet is essentially a step function, *i.e.* of the form

$u_T(t) = 0$ for $t \leq T$ and 1 for $t > T$, where T is a threshold representing the desired delay between arrival to the network and decoding at the destination.[3] The average cost (assuming voice traffic only) is then

$$(1/N) \sum_{n=1}^{N} u_T(s_n), \tag{2.5}$$

which is the fraction of packets for which $s_n > T$.

Now in the context of an operating network, a desirable control mechanism would be aware of this threshold, and would attempt to ensure that voice packets do not cross it. But we can accomplish the same effect with a convex, non-decreasing function such as c(voice,) of Fig. 1. The price paid for using the convex delay cost is that the control mechanism will continue needlessly to pay attention to packets past the threshold. However, under normal operating conditions (*i.e.* when the traffic load is not excessive), we would expect the fraction of packets exceeding the threshold to be very small if the control mechanism is effective.

As a second application, consider the access of a remote facility using a display terminal, in terms of the round-trip response time. We assume here that processing in the remote facility normally has a relatively small time requirement. Sufficiently short response times are not objectionable (or even noticeable) to the typical user and the delay cost will be flat over this range. The typical user will then become increasingly impatient with increasing response time beyond a certain point. Finally, the user resigns him or herself to the likelihood of an extraordinary delay, at which point the delay cost

[3]User satisfaction will of course depend also on the value of the threshold T itself, but the selection of a threshold is a separate issue.

[4]This functional form for delay costs is also similar to that depicted by Lawler [1964; Fig. 1] in a job shop context.

levels off again. A reasonable delay cost might look something like the dashed curve[4] of Fig. 2. The control mechanism should presumably try to keep the majority of packets below the "knee" of the curve at T(disp), and all but a negligible fraction below the "shoulder" at T_s(disp). Again, we can achieve a similar effect with the solid version c(disp,) of the curve, which is convex and nondecreasing, with the same implications as in the previous example.

In this case, the delay cost was naturally expressed in terms of the round-trip response time. To conform with our earlier definition, we might now try to find functions c(user,) and c(reply,), reflecting delay costs for individual user and reply packets, which satisfy

$$c(\mathrm{disp}, t+s) = c(\mathrm{user}, t) + c(\mathrm{reply}, s) \qquad (2.6)$$

for all t and s; but it isn't hard to show that, because c(disp,) is nonlinear, this is impossible. One could, of course, seek a "best approximation" (*e.g.* in the sense of expected values in the equation above), and intuitively we expect that any reasonable cost functions so derived will also be CND. At the same time, however, the approach that will be developed here is in principle appli-

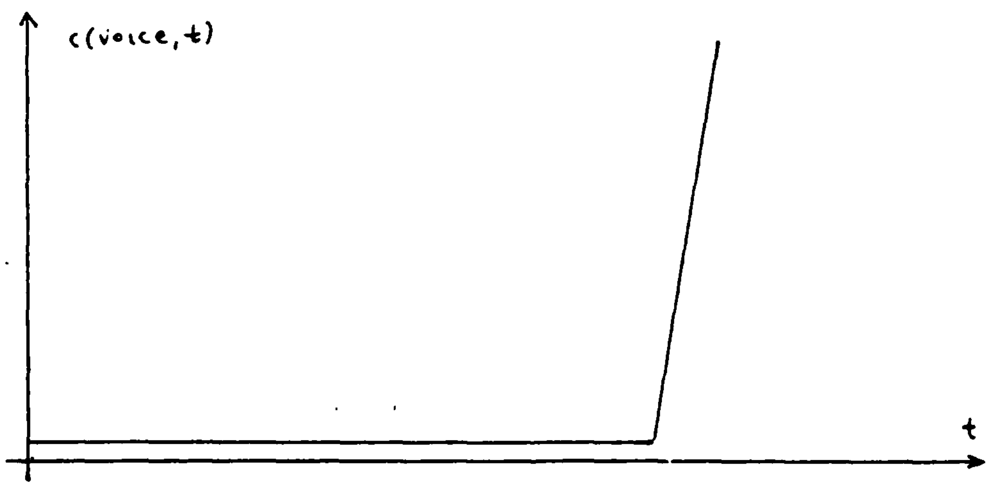

Figure 1 Possible delay cost function for packetized voice

cable to round-trip delay costs as well; the assumption of valid individual packet delay costs for "round-trip-sensitive" applications may be viewed as essentially a notational simplification.

For job-sensitive applications, the dominant consideration is the total job duration, which is in general the sum Σt_i of an unspecified number of components. The concern of the user (or manager) will typically be the fact that processing facilities and possibly personnel must be dedicated to the job for its duration; there may also be connect-time charges, both for the network and for remote resources not owned by the user. For a given application, the natural approach would again be to express these losses as a job delay cost function c(job,). In this case, however, connect-time charges will typically be linear, and in general a linear cost seems most reasonable for dedicated user-owned resources; the well-known business maxim "time is money" is offered as a partial justification. Then we can write

$$c(\text{job}, \Sigma t_i) = m\Sigma t_i = \Sigma m t_i, \tag{2.7}$$

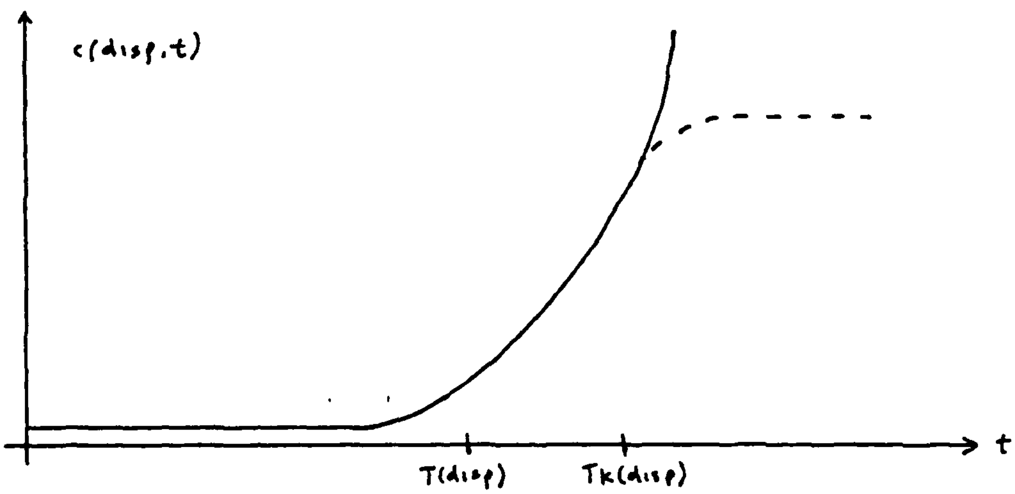

Figure 2 Possible delay cost function for display terminal

so that a linear cost $c(i,t) = mt$ (which is convex and nondecreasing) is appropriate for the individual packet delay components as well.

2.2 Stochastic Performance Measures

Of course the class b_n and system time s_n of packet n are not known in advance. In queueing and scheduling literature it is typical to model the uncertainty about such a situation by a stochastic sequence, say $\{z_n, n \in \mathbb{N}^+\}$ or $\{z_n, n=1,\dots,N\}$, where z_n is a random "vector" (or more properly a random ordered set), *e.g.* $\langle a_n, b_n, \ell_n \rangle$, whose elements represent the essential characteristics of packet n, *e.g.* the epoch[3] a_n of arrival to the system, class b_n, and length ℓ_n. The r.v. z_n are all functions on an appropriate sample space Ω, taking values in some space Z, *e.g.* $\mathbb{R}\times B\times(0,\infty)$; we assume in particular that the arrival epochs satisfy $a_n(\omega) \le a_{n+1}(\omega)$ for all $\omega\in\Omega$. The statistics of the packet sequence $\{z_n\}$ are specified by a probability measure P, which reflects the likelihood of various events defined in terms of the z_n.

Associated with each point $\omega\in\Omega$ is a sample sequence or realization $\{z_n(\omega)\}$ of the packet generation process; we may then assign to every such realization a value $x(\omega)$ from some space X, in which case the function $x:\Omega\to X$ is a random variable[4] on Ω as well, and its statistics are in principle determined by the probability measure P. We generally abbreviate events such as $\{\omega\in\Omega: x(\omega) \in A\}$ by $\{x(\omega)\in A\}$ or just $\{x\in A\}$, and if $P\{x\in A\} = 1$, we may write "$x\in A$ w.p.1" (*i.e.* with probability one).

---

[3] In deference to tradition, a "time" will denote the duration of a (temporal) event, whereas an "epoch" will denote a specific instant.

[4] A rigorous construction would require that x be measurable on Ω [Ash; §§1.5, 5.6]; this ensures probability assignments for events defined in terms of x. We will simply assume that the packet process and system parameters are such that this requirement is met for all functions on Ω defined here.

We assume that the system will treat packets in accordance with some deterministic *policy* π (chosen from a given set Π of policies), so that associated with each realization $\{z_n(\omega)\}$ of the external packet process will be a well-defined (although possibly unbounded) sequence $\{s_n(\omega,\pi)\}$ of nonnegative system times; *i.e.* the $s_n(\pi)$ are random variables on Ω. Again, obvious performance measures for the policy π are the expected total cost

$$J_N(\pi) = \sum_{n=1}^{N} E\{c[b_n, s_n(\pi)]\} \tag{2.8}$$

and the long-run expected average cost[7]

$$J(\pi) = \lim_{N\uparrow} (1/N) \sum_{n=1}^{N} E\{c[b_n, s_n(\pi)]\}; \tag{2.9}$$

when the limit exists; thus we seek practical policies for which J_N or J is small. We will not always make explicit the π dependence, but some underlying policy is always assumed.

For certain traffic classes, particularly packet-sensitive classes, it is appropriate to define performance *specifications* as well. We may construct such a specification for any given class $b \in B$ by imposing a simple constraint on the associated long-run cost. We will assume here that the event

$$\Omega_b = \{\omega\text{: } \{z_n(\omega)\} \text{ contains infinitely many class-}b\text{ packets}\} \tag{2.10}$$

---

[7]It might seem more natural to consider instead the expectation of the limit $E\{\lim (1/N)\Sigma_N c[b_n,s_n(\pi)]\}$, but (2.9) is more convenient from a technical standpoint; in any case, if there exists a random variable M such that $(1/N)\Sigma_N c[b_n(\omega),s_n(\omega,\pi)] \le M(\omega)$ for all ω, N, then by the dominated convergence theorem [Ash; p. 49] the two performance measures are equivalent for a given policy π.

[8]We can write $P(\Omega-\Omega_b) = P(\liminf_n \{b_n \neq b\})$; this will be 0 if, for instance, the b_n are independent and $\liminf_n P\{b_n=b\} > 0$ [*cf.* Ash *et al*, 1975; p. 136].

has probability one;[8] then for every $\omega \in \Omega_b$, let

$$n_j(b,\omega) \equiv \min\{N: \sum_{n=1}^{N} \chi\{b_n(\omega)=b\} = j\} \tag{2.11}$$

be the index of the jth class-b packet in $\{z_n(\omega)\}$, where the indicator $\chi\{b_n=b\} \equiv 1$ if $b_n(\omega) = b$ and 0 otherwise. Then we can require of the long-run expected average cost per class-b packet that

$$J(b) \equiv \lim_{N\uparrow} (1/N) \sum_{n=1}^{N} E\{c[b,s(n_j)] \mid \Omega_b\} \leq c[b,T(b)] \tag{2.12}$$

for some appropriately chosen "target time" T(b), such as T(disp) in Fig. 2 (thus two traffic classes may have the same delay cost but different specifications). The additive nature of the performance measures suggests that the delay costs be individually scaled (multiplicatively) such that $c[b,T(b)]$ is a constant (say 1) independent of b; this makes precise the manner in which the delay costs should reflect the relative urgency of their respective classes.

One might also want to define one or more "floating" traffic classes without specifications. In the network synthesis problem [*cf.* Gerla *et al*, 1977] performance for these classes could be traded off for savings in dollar cost, while specifications for the other classes are guaranteed. In an operational network, the cost functions for the floating classes could be multiplicatively scaled as a group, from time to time as necessary, such that specifications for the constrained classes are just met.

Finally, it will also be useful to have a performance measure in terms of the equilibrium distribution of system time associated with a given class of packets; the equilibrium distribution is a much more intuitive entity to think about, and is known for many representative queueing situations. Define

$$F_j(t;b) \equiv P\{s[n_j(b,\omega)] \le t \mid \Omega_b\}, \tag{2.13}$$

the distribution of system time for the jth class-b packet, where $n_j(b,\omega)$ is defined in (2.11). Now suppose there exists a distribution function $F(\ ;b)$ which is, in some sense, an arbitrarily close approximation to $F_j(\ ;b)$ for sufficiently large j; then $F(\ ;b)$ corresponds to what we would think of as the distribution of system times seen by class-b packets in equilibrium. The sense in which $F_j(\ ;b)$ is required to approach $F(\ ;b)$ in this context is called *weak convergence* [Ash; §8.1]; weak convergence of a sequence of probability distribution functions, denoted here by $F_n \xrightarrow{w} F$, is equivalent* to pointwise convergence $F_n(t) \to F(t)$ for every continuity point t of F (including $\pm\infty$).

Now if $F_j(\ ;b) \xrightarrow{w} F(\ ;b)$, we might expect that, in some cases,

$$\int c(b,t)dF(t;b) = J(b), \tag{2.14}$$

i.e. that the expected cost of every class-b packet arriving after the system is in equilibrium is indicative of the long-run average cost. In the case of linear costs, certain system models have been shown to obey relations similar to (2.14) [Fife] [Stidham; p. 1122]. Without assuming a specific model, we can show that (2.14) holds for every continuous, nondecreasing, nonnegative delay cost $c(b,\ )$, under a technical condition on the sequence $\{c[b,s(n_j)]\}$ which should be satisfied for most models of interest (note that every convex delay cost is continuous); a formal statement and proof are given in §A.1.

---

*however, it is often defined as follows: $F_n \xrightarrow{w} F$ iff $\int x(t)dF_n(t) \to \int x(t)dF(t)$ for every bounded, continuous function $x: \mathbb{R} \to \mathbb{R}$.

2.3 Delay Performance in Equilibrium

In the light of the previous paragraph, we expect that delay performance in equilibrium will be representative of long-run average performance as well, and we will often tacitly assume that (2.14) is true. We first obtain a characterization of equilibrium delay performance for a typical packet switching network.

2.3.1 Packet Network Model

Our model of a packet network is a set V of *nodes,* which represent packet switches, and a set E of *links,* where link <i,j> represents a one-way transmission channel from node i to node j; thus <V,E> is a (directed) graph [Lawler, 1976; §2.3]. Associated with each link is a packet queue feeding a transmitter, and we assume unlimited buffer space. The transmitter on link e has capacity θ_e bits/sec, so that transmission of a packet of random length ℓ bits on link e takes a random time $\tau_e = \ell/\theta_e$ seconds. The structure just described is often called the "subnet;" the user's computer system, or "host," is interfaced to one of the subnet nodes [Tanenbaum; §1.2]. We view the packet process $\{z_n\}$ as being generated by the ensemble of external hosts, and the policy π as a responsibility of the subnet.

Packet n is generated with a specified destination, and in real life the subnet must select for it a *route* r_n, *i.e.* a sequence of links from the origin node to the destination node. The subnet will often have other responsibilities as well, such as congestion control [Gerla *et al*, 1980]. Here, however, we want to focus on subnet scheduling; we first of all want as simple as possible a structure for the subnet policy in our model, so we now make the explicit qualification that π is a subnet *scheduling policy* only.

Secondly, we are not going to pursue in any depth the effects of possible interactions between scheduling and other subnet functions. As an exam-

ple, define $\pi_0 \equiv$ "first-come-first-served (FCFS) scheduling on every link," and suppose π_r is the same except that it always gives priority to packets on some route r. If the route assignments are identical in each case, it is intuitively obvious (and very likely true as well) that the delay on route r will be smaller under π_r than under π_0. For this reason, however, an adaptive routing strategy might assign a larger portion of traffic to r under π_r than under π_0 [Tanenbaum; §5.2].

For concreteness, we will assume that every packet vector $z_n(\omega)$ contains a route $r_n(\omega)$ in some set R; the system time $s_n(\omega)$ is then the sum of the queueing and transmission times for packet n as it traverses the succesive links in the route $r_n(\omega)$. We also implicitly assume a flow control mechanism, situated between the hosts and the subnet, which keeps link loading at a reasonable level.

Suppose now that, as $n \to \infty$, and for every route $r \epsilon R$,

$$P\{s_n \le t,\ r_n = r\} \xrightarrow{w} H(t,r). \tag{2.15}$$

It then follows that, as $n \to \infty$,

$$P\{r_n = r\} = P\{s_n \le \infty,\ r_n = r\} \to H(\infty,r) \equiv G(r), \tag{2.16}$$

$$P\{s_n \le t\} = \sum_R P\{s_n \le t,\ r_n = r\} \xrightarrow{w} \sum_R H(t,r) \equiv F(t), \tag{2.17}$$

and, as long as $P\{r_n = r\}, G(r) \neq 0$,

$$P\{s_n \le t \mid r_n = r\} = P\{s_n \le t,\ r_n = r\} \div P\{r_n = r\} \xrightarrow{w} H(t,r) \div G(r) \equiv F^r(t). \tag{2.18}$$

Then $F(t) = \Sigma_R F^r(t) G(r)$, *i.e.* the equilibrium distribution of system times is the sum of the distributions corresponding to the individual routes weighted by the equilibrium route probabilities, as we might have expected.

By making some simplifying assumptions, we can invoke a standard result to get a qualitative picture of $F(t)$. It is typical to model a packet network as a collection of independent M/G/1 queues, because the distribution of delay is known in this case [Kleinrock, 1975; ch. 5]. In particular, for an M/M/1 queue in which packets are transmitted FCFS,[18] the density of delay (queueing plus transmission) on link e has the exponential form $f^e(t) = (\mu_e-\lambda_e)\exp[-(\mu_e-\lambda_e)t]$, where $\mu_e = 1/E(\tau_e)$ is the service rate and λ_e is the arrival rate [Kleinrock, 1975; p. 202]. Moreover, if the link capacities θ_e are such that $\delta_e \equiv (\mu_e-\lambda_e)$ is independent of e, the system time over a given route r of j hops (links) has an Erlang density

$$f^r(t) \equiv \partial F^r(t)/\partial t = \delta^j t^{j-1}\exp(-\delta t)\div(j-1)!, \tag{2.19}$$

expectation $S^r \equiv \int t dF^r(t) = j/\delta$, and variance j/δ^2 [for a picture see Kleinrock, 1975; Fig. 4.5, p. 124]; thus for such an idealized model the overall equilibrium density $f(t) = \Sigma_R f^r(t)G(r)$ of system times is a weighted sum of the Erlang densities corresponding to the different routes, and might look something like Fig. 3. While the M/M/1 packet network model is a fairly reasonable one from the standpoint of mean delay [Kleinrock, 1976; ch. 5], we do not expect an accurate representation of the entire distribution; yet neither do we expect such limitations to materially alter the qualitative picture developed here.

---

[18]This model is equivalent to the "network of Markovian queues" [Kleinrock, 1975; §4.8] (in which case the queues correspond to the vertices of a graph rather than the edges).

2.3.2 Implications of Convex Delay Costs

For a linear cost $c(t) = mt$, we don't really care about the shape of f, we just want the equilibrium mean $S \equiv \int t dF(t)$ to be small. However, as the convexity of c becomes more pronounced, the mean becomes less crucial, and we generally become concerned more with higher moments of delay; *e.g.* if $c(t) = (t/T)^k$ (T is some threshold), we want the kth moment $\int t^k dF(t)$ to be small. In turn, the higher moments are typically driven by the upper tail of the equilibrium density, which corresponds to packets whose routes have large numbers of hops,[11] and/or packets which have long queueing times at some links on their routes. Transmission scheduling offers the potential for mitigating both of these contributions to the upper tail, by giving priority (at the transmitter) to packets whose routes have large numbers of hops, and by giving priority to packets which have had long queueing times at "upstream"

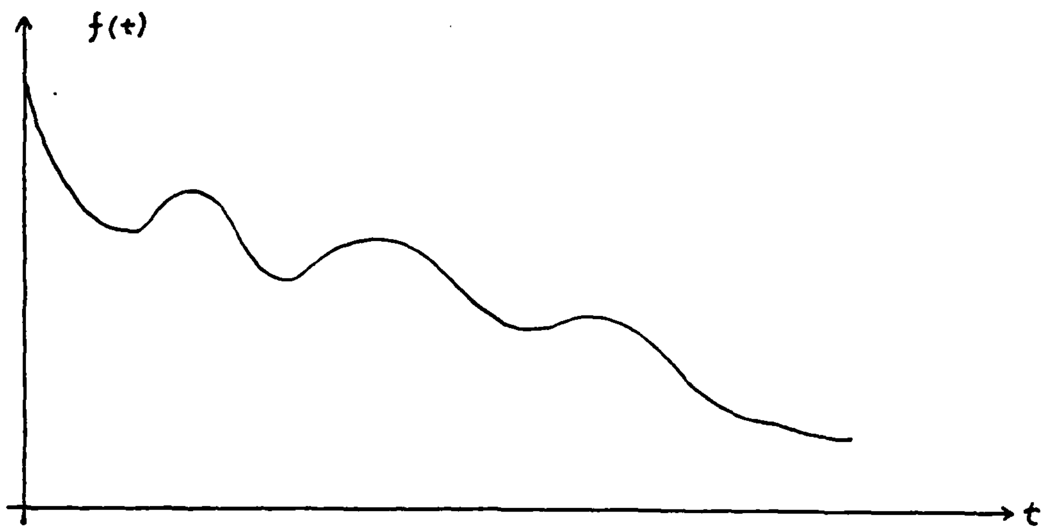

Figure 3 Density of system time in a hypothetical M/M/1 network

[11] *e.g.* in the ARPANET c. 1976, many node pairs had routes with as many as 7 hops [Gerla *et al*, 1977; Fig. 1, p. 49].

(*i.e.* earlier) links in their route, or are expected to do so at "downstream" (*i.e.* later) links.

We immediately qualify this, however, by noting that, in order for scheduling to be effective, there must be packets to schedule. We should expect that scheduling can be most effective in environments charcterized by large numbers of packets in queue, such as high-speed systems. Scheduling might also turn out to be most useful as a smoothing response to transient load surges.

We expect that pulling the upper tail of the equilibrium distribution in toward the mean must come at the expense of pulling the lower "tail" in toward the mean as well, *i.e.* we expect some sort of conservation of the mean system time S. In a single work-conserving G/G/1 queue the mean is conserved iff packets are selected independent of their transmission times [Kleinrock, 1976; §3.4] [Heyman *et al*, 1982; §11.5]. We expect the situation to be a little more complicated in the network case, but the single queue derivation suggests that the analogous quantity is the *sum of transmission times* on the given link *and on downstream links*. Thus if we give priority to packets with many hops, we might anticipate an increase in mean delay as well as a reduction in higher moments. By Little's formula [*cf.* Stidham, 1972; pp. 1122-23] this would increase the expected number in system, which is bad from the standpoint of congestion; but by generalizations of Little's formula [Brumelle] [*cf.* Kleinrock, 1975], higher moments of number in system should be reduced, which seems desirable from a congestion standpoint.

Insofar as the mean system time S is conserved, we can alternatively view convex costs as being generally driven by the central moments $\int |t-S|^k dF(t)$ of equilibrium delay, such as the variance. Indeed, Jensen's *inequality* [Ash; p. 287] gives $\int c(t)dF(t) \geq c(S)$ for every convex function c; moreover, this holds with equality if and only if the distribution F is a unit

step (*i.e.* the density f is a unit impulse) at the mean S. Thus given a collection of distributions with the same mean, those which best approximate (in some sense) a unit step are the most desirable, from the standpoint of any given convex cost; this is essentially a restatement of the basic idea of §2.1.

The variance of equilibrium delay similarly has components due to the variance of route hop-length and the variance of delay at individual links (or along individual routes). Wong *et al.* have recently addressed the effect of route hop-length variance with packet scheduling: they derived a time-dependent scheduling discipline (*cf.* §3.3.2), based on route membership only, that minimizes the variance $\Sigma_R(S^r-S)^2G(r)$ of *mean* route delays in equilibrium. That work was motivated by a concern for fairness among different node pairs (*cf.* §1.2), and the scheme effectively aligns the means of the densities $f^r(t)$, subject to certain connectivity constraints. However, the effect on *the higher moments of delay along individual routes* is not apparent. We also note that their simulation results showed little variation in mean delay due to scheduling discipline.

The convexity property also suggests an approach to lower bounds on the expected cost. Denote by $F(\ ;\pi)$ the equilibrium distribution under policy π, and define $S(\pi) \equiv \int t dF(t;\pi)$. If we can find a policy π^* such that $S(\pi^*) \le S(\pi)$ for all $\pi \epsilon \Pi$, then for every convex and nondecreasing function c, we have

$$c[S(\pi^*)] \le c[S(\pi)] \le \int c(t)dF(t;\pi) \tag{2.20}$$

by monotonicity and Jensen's inequality. For a single M/G/1 system, it is well known that the desired π^* is that policy which gives priority at each departure instant to a waiting packet with the smallest transmission time (see §3.2.2). From the discussion above, a network analogue to the optimal single queue policy should suggest itself. However, we should not expect such bounds to be very tight, because π^* *is not likely to achieve equality in the second*

inequality. Moreover, the idea does not seem to generalize to the case of multiple packet classes in a useful way.

Finally, we mention the possibility of improving delay performance through selective packet routing. There are a great many approaches to routing [Tanenbaum; §5.2], but the effective goal can be expressed as follows: for each origin-destination pair of nodes, the composite traffic flow rate (in bits/sec) required by the associated users is apportioned among some designated set of paths; the objective is a profile of link flows that gives rise to a low expected path delay per packet. Given such a flow assignment, then, one could reserve the lower delay paths to a given destination for the more urgent traffic. However, the examples given in §A.2 suggest that, for larger utilizations, such a scheme will not in general be as flexible as scheduling.

CHAPTER III - SCHEDULING ALGORITHMS FOR PACKET NETWORKS

Given the network configuration, the packet generation process, and the delay costs, we want a scheduling policy π to reduce or minimize the performance measure $J(\pi)$ or $J_N(\pi)$. A scheduling policy basically specifies, for each departure instant, the next packet to be transmitted. An urgent, newly-arrived packet might also be allowed to displace or "pre-empt" the one in transmission.[12] Without allowing pre-emption, one might still decide to leave the transmitter idle after a departure if a relatively urgent packet were expected in a very short while. On the other hand, both pre-emption and inserted idle time typically complicate queueing system analysis as well as increase the expected delay, and pre-emption has associated protocol problems in packet networks; for these reasons we will not consider such strategies here.

3.1 A Decentralized Scheduling Problem

The network scheduling problem arises in a number of other contexts, for example the so-called "job shop" scenario, in which pieces of work must be processed on a number of different machines, perhaps not in the same order. [see *e.g.* Conway *et al*]. In many such cases, the time-scale on which decisions are made is relatively long; it is then reasonable to assume that, for a given realization ω, the characteristics $z_n(\omega) = \langle a_n(\omega), b_n(\omega), \ell_n(\omega), r_n(\omega)\rangle$ of arrival n (or estimates thereof) become known to a single scheduler at the epoch $a_n(\omega)$.

In a packet network, however, this information becomes available at $a_n(\omega)$ only at the first link in the route $r_n(\omega)$. Collecting such detailed state informa-

[12] For analysis of some pre-emptive queueing policies see *e.g.* Conway *et al*. [§8.7] and Schrage [1968].

tion for a global scheduler (and passing instructions back to the individual nodes) is clearly out of the question; at the same time, however, most of this dynamic information is of relatively little value in making scheduling decisions for a given link. This motivates consideration of distributed schemes, in which each link makes its own scheduling decisions, based on a combination of locally available information and limited communication with other nodes.[13] A reasonable performance measure for such a link scheduler would then be the expected (average) cost of all packets transiting that link; such a scheduler will exert a minimal influence on other packets, which would in any case be difficult to quantify. Thus π will be interpreted as a *link* scheduling policy, to be executed independently at every link.

Specifically, we will consider implementations characterized by the following assumptions about the kinds of information available to the link scheduler:

- The delay cost is known for every class; we do not antipicate a large number of classes in practice.

- Certain sample statistics are known about delays on the routes in which the link is included; the use of these is discussed in §3.2.

Furthermore, for each transiting packet:

- The epoch of arrival to the link itself is of course known.

- The traffic class is known.

- The delay at upstream links in the route (or equivalently, the epoch of arrival to the system) is known; this has obvious practical implications, which are discussed below.

---

[13]Such an approach might be desirable as well in certain applications with longer time scales, as an alternative to more complicated global strategies.

- The *route* is known; actually, we do not need the route *per se* (which may not be known in practice), just some key to the delay statistics associated with the route, such as the session identifier [Tanenbaum; §§5.4.1, 5.4.4].

- The transmission time on the link (or equivalently the length) is easily established during arrival.

Given $\{z_n\}$ and π, the packet process seen by a given link is specified; however, it will certainly be very "messy," even if $\{z_n\}$ is "nice." In fact, the link arrival process will in general have a mild dependence on the link departure process, and hence on the link scheduling policy, through feedback paths. To avoid these difficulties, we will simply define a new link process $\{y_n\}$ from scratch, with the assumption that the statistics relevant to the scheduling problem are similar to those that would arise under the old network process $\{z_n\}$. The elements of y_n will be: the epoch a_n of arrival *to the link;* the class b_n; the "history" h_n, *i.e.* the time behind packet n in the system; the route r_n; and the transmission time τ_n. P will now denote the probability measure for the link process.[14] It is in this case useful to make the explicit definitions

$$y_n(\omega) \equiv \langle a_n(\omega), b_n(\omega), h_n(\omega), r_n(\omega), \tau_n(\omega)\rangle \in \mathrm{IR}\times B\times [0,\infty)\times R\times(0,\infty) \equiv Y \qquad (3.1)$$

for every $\omega\in\Omega$.

The assumption of accurate knowledge of each packets history is an important one here. For example, priority schemes based on number of hops in route, and number of hops remaining in route, were considered, but often had much worse second moment properties than FCFS. The obvious way to

[14] In principle, knowledge of the statistics of $\{y_n\}$ might help the link make scheduling decisions, but we will not consider this approach here.

[15] Thus no additional effort is entailed by a round-trip-sensitive reply packet retaining the history from the user leg of its trip (*cf.* §2.1).

communicate the packet history is for every packet to contain a bit field, which is either "stamped" once with the epoch of arrival to the system, or updated by each node in the route to reflect the new history.[18] Thus there is an obvious tradeoff between resolution and link loading, and the assumption of precise information can never really be satisfied. On the other hand, we will see that histories are not really necessary for packets with linear costs. In other cases there may be sneaky ways to circumvent the requirement as well: for instance, in a packetized voice protocol, a precise time stamp may be forwarded at the onset of every session (or talkspurt, as the case may be); the system-arrival epochs of subsequent packets can then be recovered from their sequence numbers if all nodes have accurate clocks. However, the relationship between resolution and performance is certainly not clear and requires further attention.

3.2 A Dynamic Programming Formulation of Link Scheduling

The behavior of a queueing system may often be modeled as a continuous-time Markov process [see *e.g.* Heyman *et al*, 1982; chs. 8, 9]. In many cases, however, we are interested in the system behavior only at a certain (countable) set of epochs, and the behavior on this "embedded" set of epochs is often a discrete-time Markov process with a simpler state space than that of the underlying process; a standard example is the M/G/1 queue [Kleinrock, 1975; §5.3]. We will construct such a Markov model for the link scheduling problem, or more precisely a model "schema" or template. The model is essentially a dynamic programming formulation [Bertsekas, 1976; chs. 2, 9]; it is also similar to the Markov decision process [Heyman *et al*, 1984; ch. 4]. For other applications of dynamic programming and Markov decision processes to queueing system control see *e.g.* Stidham *et al*. [1974] and references therein.

3.2.1 Process State and Dynamics

We first need a notion of process state. We want to make a scheduling decision whenever the transmitter is idle and faces a non-empty queue; we will represent the kth such *decision epoch* by t_k, $k = 0, 1, \ldots$.

We want also to represent the characteristics of the packets in queue. Let Q be a subset of $\mathbb{N}^+$, and ν a function from Q into Y; the function ν can be interpreted simply as a list of packets and their characteristics; indeed, the rigorous definition of a function $\nu: Q \to Y$ is the *set of ordered pairs* $\{<n,y>: n \in Q, y = \nu(n)\}$ [see *e.g.* Suppes; ch. 3]. Then if Υ is the set of all such lists,[16] we may represent by $\nu_k: \Omega \to \Upsilon$ the *queue list*, and by Q_k the set of packets in queue, at the decision epoch t_k.

Finally, in order that the process be Markov, we typically need to record some additional information about the process history up to epoch t_k. We denote this information by $\psi_k: \Omega \to \Psi$, where Ψ is an appropriate space; the nature of the ψ_k will be elaborated on below. Thus we define the state representation x_k at the kth decision epoch, and the state space X, by

$$x_k(\omega) \equiv <t_k(\omega), \nu_k(\omega), \psi_k(\omega)> \in \mathbb{R} \times \Upsilon \times \Psi \equiv X; \qquad (3.2)$$

note that, given $\nu \in \Upsilon$, the corresponding packet-set Q is always determined as $\{n: <n,y> \in \nu \text{ for some } y \in Y\}$, *i.e.* as the domain of ν. In particular, the initial state x_0 will include components defined by

$$t_0 \equiv a_1, \qquad (3.3)$$

$$Q_0 \equiv \{n: a_n = a_1\}, \qquad (3.4)$$

---

[16] Note that every such function $\nu: Q \to Y$ is a subset of $\mathbb{N}^+ \times Y \equiv \{<n,y>: n \in \mathbb{N}^+, y \in Y\}$ (think of the graph of ν in $\mathbb{N}^+$-Y coordinates); then Υ is the space of all subsets of $\mathbb{N}^+ \times Y$ which are also functions. Other "big" sets defined here (such as Π) can be precisely specified in a similar manner.

$$\pi_0 \equiv \{<n,y_n>: n\epsilon Q_0\}. \tag{3.5}$$

We now represent the dynamics which govern the motion of the process through the state space. We will say a map μ:X→IN$^+$ is *admissible* if $\mu(x)\epsilon Q$ for all $x\epsilon X$. Then we formally define (at last) an admissible scheduling policy as a sequence $\pi \equiv <\mu_0, \mu_1, \ldots>$ of admissible maps on the state space; let Π be the set of all such policies. The interpretation is that, at the epoch t_k, the scheduler selects for transmission the packet

$$u_k \equiv \mu_k(x_k) \;\epsilon\; Q_k \tag{3.6}$$

(in the finite case only the first N maps are used); packet u_k subsequently departs at epoch $d(u_k) \equiv t_k + \tau(u_k)$. A policy $\pi\epsilon\Pi$ is *stationary* if $\pi = <\mu, \mu, \ldots>$ for some μ.

The next decision epoch can be written as

$$t_{k+1} \equiv \begin{cases} a_{k+1} & \text{if} \quad Q_k - u_k = \emptyset \quad \& \quad a_{k+1} > d(u_k), \\ d(u_k) & \text{otherwise} \end{cases} \tag{3.7}$$

where $\emptyset$ is the empty set. Let $E_{k+1} \equiv \{n: t_k < a_n \le t_{k+1}\}$ be the set of packets which enter the queue in the interval following the kth decision; then

$$Q_{k+1} \equiv Q_k - u_k \cup E_k, \tag{3.8}$$

and the new queue list is given by

$$\pi_{k+1} \equiv \{<n,y_n>: n\epsilon Q_{k+1}\}. \tag{3.9}$$

For a given policy π, denote by $\{x_k(\pi)\}$ the process which results from the sequence of decisions $\{u_k = \mu_k(x_k), k\geq 0\}$.

As a concrete example of how the μ_k would be chosen, suppose that $\{y_n\}$ is the superposition of independent processes for each $b\epsilon B$, and that for

each component process, the interarrival times, histories, routes, and transmission times are i.i.d. and independent of each other; then it is suffucient that ψ_k includes for every class the latest arrival epoch up to and including t_k. On the other hand, a model of this type may in principle be constructed for the system behavior under completely arbitrary packet processes, by recording the entire state history in the ψ_k.

3.2.2 Decision Costs

The process $\{y_n\}$ says nothing about the probabilistic behavior of packets after they leave the link. As the basis of a performance measure for the link scheduling problem, we will use the delay statistics (and the delay costs) to develop a *decision cost* $g: \mathbb{R} \times Y \to [0,\infty)$; here $g(t,y)$ will represent an *estimate* of the cost (on system exit) for a packet with characteristics y which begins transmission at epoch t on a given link. Thus $g[t_k, y(u_k)]$ will represent the estimated cost for the packet u_k selected at epoch t_k, and we may redefine J_N and J as (expected) estimated costs of using the policy π by[17]

$$J_N(\pi) \equiv \sum_{k=0}^{N-1} E\{g[t_k(\pi), y(u_k)]\}, \tag{3.10}$$

$$J(\pi) \equiv \lim_{N\uparrow} (1/N) \sum_{k=0}^{N-1} E\{g[t_k(\pi), y(u_k)]\}, \tag{3.11}$$

where $y(u_k)$ abbreviates $y\{\mu_k[x_k(\pi)]\}$. It is not hard to show that (2.8) and (3.10) are equivalent, in the sense that they would have the same value in a single link system; assuming convergence, we might expect the same of (2.9)

---

[17]It is typical in dynamic programming to condition the performance measure on a given initial state; in the present context we expect any such dependence to be very mild at worst and we will not worry about it explicitly.

and (3.11), although a proof is not attempted here. The link scheduling problem is then effectively specified by a probability measure and decision cost <P,g>; given such an instance of the problem, a policy which minimizes (3.10) or (3.11) will be called optimal.

Our approach to the decision cost g at a given link is to construct an estimate f(t,y) of the "future," *i.e.* of the delay on downstream links, for a packet with characteristics y selected at t; we then define

$$g(t,y) \equiv c\,[b,\ h + (t-a) + \tau + f]. \tag{3.12}$$

Of course the actual packet futures must have some variation about the estimate f, and for strictly convex delay costs this will drive up the average cost beyond the estimate given by (3.12). In certain such cases, the bias is irrelevant to the optimization problem: for instance, if $c(b,t) = t^2$, then $g(t,y) = (h+t-a+\tau+f)^2 = t^2 + 2(h-a+\tau+f)t + (h-a+\tau+f)^2$, and the f^2 term is an isolated constant. For a cubic delay cost, however, the slope of the decision cost would be biased by an f^2 term (in this case one might consider the additional use of an estimate of the second moment of downstream delay). As the convexity becomes more pronounced, however, we should also expect that an effective scheduling algorithm will tend to reduce the variation in packet futures for any given <t,y>.

Thus the link essentially interprets the downstream estimate f as a delay which has already been incurred; it is then not completely unreasonable to define the *effective elapsed time*

$$\eta(t,y) \equiv h + (t-a) + f(t,y) \tag{3.13}$$

of a packet with characteristics y at epoch t. The link is thus assumed to be already "liable" for the amount $c\,[b,\eta(t,y)]$ at epoch t, and the decision cost

may now be expressed simply in terms of the effective elapsed time as $g(t,y) = c(b,\eta+\tau)$.

To construct our future estimate f, we assume that the link scheduler has a table which gives an estimate $D(b,r)$ of the delay to the destination on downstream links for a class-b packet on route r; of course $D(b,r) = 0$ if the link is the last one on route r. In practice, the estimates are likely to be sample moving averages of the form

$$(M-m)^{-1} \sum_{j=m}^{M-1} D_j(b,r,\omega,\pi), \tag{3.14}$$

where $D_j(b,r,\omega,\pi)$ is the downstream delay of the jth class-b packet on route r to reach the destination. The table would be updated periodically using reports from the destination nodes. Adaptive routing algorithms often employ such tables of downstream delays (for a single class) [Tanenbaum; §5.2]. For simplicity, we will assume that D is independent of ω and π, since we have no intention of accounting for these complications. Thus we obtain a simple estimate of the future by setting[18]

$$f(t,y) = D(b,r). \tag{3.15}$$

Note that f is independent of t here, so that $g(\ ,y)$ is simply a shifted version of $c(b,\ )$, and has the same functional form.

[18]And this can similarly be modified to include an estimate of the system time on the return leg for a round-trip-sensitive user packet.

3.2.3 A Refinement of the Downstream Delay Estimate

For a given pair <b,r>, the goal of the scheduler may be viewed as minimizing the variation in estimated system times of transiting packets (*cf.* §2.3.2). The packet histories, arrival epochs, and transmission times will be taken into account by the downstream links, and this should help rectify imbalances in the delays accumulated at the given link and the upstream links associated with transiting routes; the estimate $f = D(b,r)$ is somewhat "pessimistic" in that is does not anticipate any such help from the downstream links. The following example illustrates how this may degrade system performance.

Suppose for a given realization there are four packets in queue, all of the same class b and with identical transmission times τ. The link is the last one in a route shared by packets 1 and 2, so their estimate is $f_{1,2} = 0$; their accumulated delays differ by 3τ seconds. The link is the first one in a route shared by packets 3 and 4, with some estimate $f_{3,4}$; they have arrived 5τ seconds apart.

This is clearly the last opportunity to do anything about the relative delays of packets 1 and 2, and in fact their system times can be made exactly equal if the schedule 1-3-4-2 is followed. This schedule also reduces by τ the difference in accumulated delays for packets 3 and 4 (as seen by the next link) to 4τ; moreover, there is a good chance that this difference can be reduced further, if not eliminated entirely, by the downstream links. But the pessimistic scheduler is in effect operating under the mistaken assumption that this is also the last opportunity to do anything about packets 3 and 4. Fig. 4 shows how the effective elapsed times η_n of the four packets might be related in the worst case; the average decision cost at the link is then minimized by the schedule 3-1-2-4 (this is easy to prove, and we do so in §3.4, but the intuition should be fairly clear). This schedule indeed reduces the

differences in accumulated delays for packets 3 and 4 to 2τ, but results as well in an irrevocable difference of 2τ in the total system times of packets 1 and 2.

We can also imagine the opposite extreme, that of the totally "optimistic" scheduler; this one assumes that the downstream links are somehow able to guarantee equal system times for all packets with the same $\langle b,r \rangle$ (a particularly poor assumption on the last link in a route). To generate such optimistic estimates of downstream delays, a link might maintain a table which gives an estimate $S(b,r)$ of the total system time for a class-b packet on route r, and derive a time-dependent estimate $f(t,y) = S(b,r) - [h + (t-a) + \tau]$. This gives $\eta(t,y) = S(b,r) - \tau$ and $g(t,y) = c[b,S(b,r)]$; with this estimate, then, decisions may be based on at most packet class and route, and cannot be very meaningful.

The example of Fig. 4 shows that a link has a limited capacity for reducing variations in accumulated delay. Thus a more "realistic" scheduler should expect that, for a given pair $\langle b,r \rangle$, each downstream link will make some contribution to this goal, and it should realize that some effort on its own

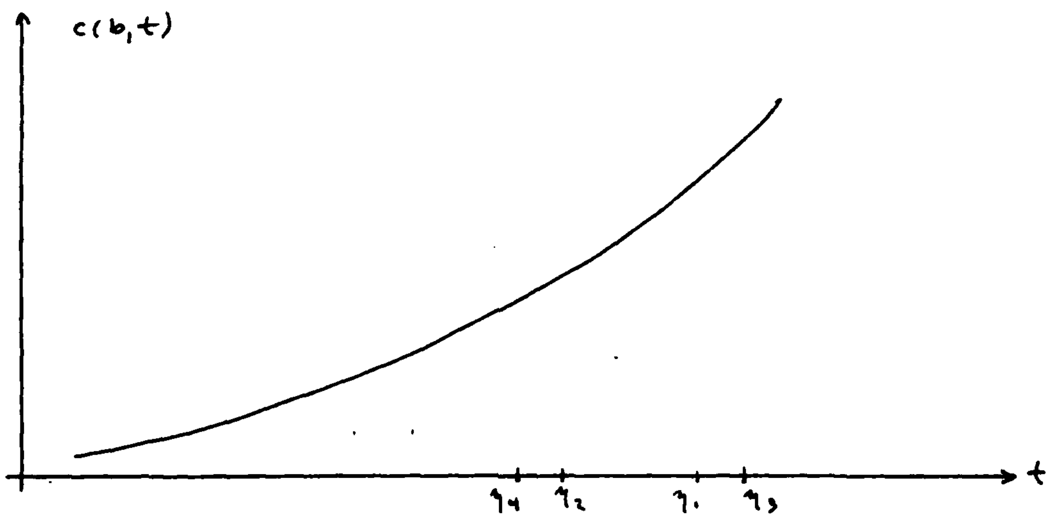

Figure 4 Elapsed times with pessimistic estimates

part will be required as well. We may strike such a balance between the pessimistic and optimistic schedulers with an estimate of the form

$$f(t,y) = \beta_y D(b,r) + (1-\beta_y)[S(b,r) - (h+t-a+\tau)] \tag{3.16}$$

where $\beta_y \epsilon (0,1)$ represents the fraction of the responsibility assumed by the link for a packet with characteristics y.

As one possibility, if j_r is the number of downstream links on route r, we might set $\beta_y = 1/(j_r+1)$. Then for packets at early links in a route, the differences in effective elapsed times are reduced to much more reasonable values than those seen by the pessimistic scheduler; however, they are not completely eliminated, so that meaningful scheduling decisions may still be made. Of course other assignments for β_y are possible, and experimentation would probably be helpful in establishing reasonable choices; but one obvious constraint is that, on the last link in a route, β_y should go to 1, so that f goes to 0.

Now, however, f is affine in t, and we find that

$$g(t,y) = c(b, \beta_y t + K_y) \tag{3.17}$$

for some K_y independent of t. While the convexity of c(b,) is retained in g(,y), the particular functional form need not be; we will see in §3.4 that the characterization of an optimal policy becomes more difficult in this case.

3.2.4 Performance on a Saturated Link

Consideration of scheduling disciplines other than FCFS opens up the possibility that some packets may get "trapped" in the system forever, and we clearly want to avoid this. Indeed, both (2.9) and (3.11) have little significance unless (with probability one) s_n is finite for every n. If this condition

is not met, then[19] $E[c(b_n, s_n)] = \infty$ for some n, so that (2.9) cannot converge. And because (3.11) accounts for packets in order of departure, it doesn't even give any indication of this problem; in fact the scheduler might conceivably minimize (3.11) while neglecting indefinitely the most urgent packets.

To ensure no trapping, we could require that on each link the queue empties out infinitely often, *i.e.* the queue busy-periods are all finite [this is typically sufficient for stability of a queue: Kleinrock, 1975; p. 93]; then every packet is selcted eventually, at every link in its route. More precisely, suppose that, on every link, we have

$$\text{(w.p.1)} \quad \liminf_{k\uparrow} |Q_k| = 1, \tag{3.18}$$

where $|Q|$ denotes the number of elements in a set Q; recall that by definition $|Q_k|$ is never 0. Thus for every k there exists a j>k such that $|Q_j| = 1$ [*cf.* Rudin; pp. 55-57]; then (w.p.1) for every packet n, we have $n \in E_k$ for some[20] k, and hence $d_n \le t_j + \tau_n$ and $s_n \le t_j - t_k + \tau_n$ for some j>k.

We *might* also imagine a situation in which the finite busy-period assumption is violated, particularly when the potential traffic is large. As a somewhat weaker assumption, then, we require only that

$$\text{(w.p.1)} \quad \liminf_{k\uparrow} |Q_k| = M < \infty \tag{3.19}$$

for some M, *i.e.* for every k there exists a j>k such that $|Q_j| \le M$; we do expect that such an M exists in practice, regardless of the policy. We may

[19] Suppose that, for every n, $E[c(b_n, s_n)]$ is finite: then by Jensen's inequality $E(s_n)$ is finite, hence s_n is finite w.p.1; then the probability of the set of realizations in which s_n is infinite for some n is $P(\cup_{n\ge1}\{s_n \text{ infinite}\}) \le \sum_{n\ge1} P\{s_n \text{ infinite}\} = 0$.

[20] this actually presumes in addition that (w.p.1) $\liminf_{n\uparrow} \tau_n > 0$.

loosely refer to a link for which $M>1$ as marginally stable (pathological instabilities exist). Trapping is now possible for some policies, which suggests (correctly) that (3.18) is also a necessary condition (for no trapping under every policy); but an argument similar to the one in the previous paragraph shows that no trapping occurs under FCFS, so that at least one such policy always exists.

We will address the marginally stable case by holding the link accountable for every arriving packet, as follows: in §3.3.1, $\Gamma^*(t,\upsilon)$ is defined as the optimal cost of emptying out the queue described by the list υ, starting at epoch t; we redefine J (for the last time) by[21]

$$J(\pi) \equiv \lim_{N\uparrow} (1/N)\, E\{\sum_{k=0}^{N-1} g[t_k(\pi), y(u_k)] + \Gamma^*[t_N(\pi), \upsilon_N(\pi)]\}. \tag{3.20}$$

$\Gamma^*(t_N, \upsilon_N)$ represents the liability of the link after N decisions (assuming no additional arrivals). Now if $(1/N)E\{\Gamma^*[t_N(\pi), \upsilon_N(\pi)]\} \to 0$, then obviously $J(\pi) = \lim_{N\uparrow}(1/N)J_n(\pi)$. Moreover, we expect intuitively that (w.p.1) no trapping occurs in this case, and that (2.9) and (3.20) are equivalent.

It would be nice if we could guarantee that (w.p.1) the optimal policy never traps in the marginally stable case, but this is not true; for instance, in the case of linear delay costs $c(b,t) = m_b t$ and constant transmission times, we will show in §3.3.2 that (in the single-queue case) packets with minimum m_b always have the lowest priority, regardless of their effective elapsed times. In principle, there should exist conditions under which the optimal policy will never trap; a likely candidate would be unbounded marginal costs for every class. In such a case, Γ^* would serve as a "penalty function" which discour-

---

[21] Thus J is no longer a true per-packet average; this allows for further anomalies, which we will not pursue, but the reader is invited to explore.

ages the scheduler from neglecting any particular packet for too long. In practice, however, we cannot guarantee such a condition. Alternatively, we could establish absolute time limits such that a packet automatically goes onto a special subqueue with preferred status once its limit has been reached; this will also guarantee no trapping.

3.3 A Class of Efficient Scheduling Policies

It is appropriate in this context to focus on simple policies which can be executed quickly. The transmission time of a data packet is typically on the order of 10 milliseconds; thus even an (inexpensive) processor, dedicated to the task of scheduling, will have time for only a few thousand machine instructions per decision.

3.3.1 Myopic Scheduling Policies

In particular, it seems reasonable first of all to forget about the dynamic aspect of the problem; *i.e.* at any given decision epoch, ignore the fact that more packets are expected to arrive in the future, and address only the static problem of emptying out the queue at minimum cost. Policies of this type will be called *myopic scheduling policies* [*cf.* Heyman *et al,* 1984; ch. 3]; the notion will not really demand a precise definition.

In general, the arrival of new packets can upset the optimality of a static schedule, as the following example shows. Consider a single link system, and suppose that for a given realization packets 1 and 2 are present, and packet 3 will arrive in τ seconds. All packets have transmission times τ. The delay costs are $c(b_1,t) \equiv \epsilon \max\{0,t-\tau\}$, $c(b_2,t) \equiv a \max\{0,t-2\tau\}$, and $c(b_3,t) \equiv a \max\{0,t-\tau\}$, with $\epsilon << a$, as shown in Fig. 5. The optimal sequence for packets 1 and 2 is clearly 1-2, which has total cost 0. But if packet 1 goes first, then one of packets 2 or 3 must incur a cost of $a\tau$, whereas the policy which

results in 2-3-1 has total cost $2\epsilon\tau < \epsilon\tau$; depending on the process statistics, then, the myopic policy need not be optimal. In certain cases, however, a myopic policy can minimize the expected cost[22] as defined in §3.2.

To formalize the static scheduling problem, let $\nu \in \Upsilon$ be a queue list with $|Q| = q$ packets. We define a *static sequence* for ν as a collection $\sigma \equiv \{\sigma_n, n \in Q\}$, where σ_n is the set of packets in Q to precede packet n. The σ_n must then be strictly nested, *i.e.*[23]

$$\emptyset = \sigma(n_1) < \ldots < \sigma(n_q) = Q - n_q \tag{3.21}$$

where n_1 is the first packet according to the sequence σ and n_q is last. Given a decision cost g, the cost of scheduling ν in accordance with σ, starting at epoch t, may then be written compactly as

$$\Gamma(\sigma,t,\nu) \equiv \sum_{n \in Q} g[t + \sum_{j \in \sigma_n} \tau_j,\ \nu(n)], \tag{3.22}$$

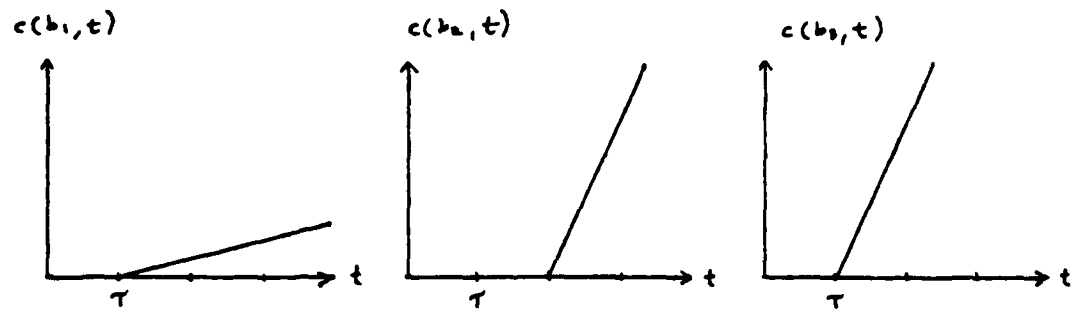

Figure 5 How myopic policies can fail

[22]There are other such sequential decision problems, *e.g.* some inventory problems, for which the optimal policies are myopic under certain conditions [see Heyman *et al*, 1984; ch. 3].

[23]Alas, "<" denotes proper set inclusion.

and this motivates our definition; but we will often revert to the shorthand $n_1 - n_2 - \ldots - n_q$. The optimal cost for the static problem <t,ν> will be defined as

$$\Gamma^*(t,\nu) \equiv \min_{\sigma} \Gamma(\sigma,t,\nu) \tag{3.23}$$

(*cf.* §3.2.2), and the sequence σ^* is optimal for the problem <t,ν> if it achieves the minimum in (3.23).

There are q! possible static sequences for a given collection of q packets; the search for more efficient solutions to this problem has been relatively active over the last twenty years. There seem to be two basic approaches, the first of which consists of devising rules for eliminating non-optimal sequences from consideration. As the most straightforward example, dynamic programming approaches can reduce the computation required in the general problem to the order of 2^q [Held *et al.*] [Schild *et al.*] [Lawler, 1964]. When the cost functions take a particular form, we can apparently do better; additional elimination criteria have been given for non-decreasing costs [Elmaghraby], CND costs [Henderson *et al*; §3], and quadratic costs [Schild *et al*]. But there is nothing to suggest that the complexity of these schemes is better than exponential, which is unacceptably slow in our context. Moreover, it is not clear that it is any less difficult to find the first item in an optimal sequence (which is all we really want for a dynamic policy) than it is to find the whole sequence.

3.3.2 Priority Rules

The second approach is the use of intuition (and a little luck) to find selection rules for the first item when the decision costs have a particularly auspicious form; these rules can typically be expressed as *priority rules,* which are (at worst) order m at each decision, and this is as good as we could reasonably have hoped for. We may interpret any function ϕ: $\mathbb{R}\times Y \to \mathbb{R}$ as a *priority function;* when a decision is to be made among the packets in a queue list ν at epoch t, we simply select a packet $n\epsilon\Omega$ for which $\phi[t,\nu(n)]$ is maximum, *i.e.* we select a packet with the highest priority [*cf.* Schrage, 1973]; for concreteness (and without loss of generality), we assume that ties are resolved in favor of the lowest index. If $\phi(t,y)$ is independent of t for all $y\epsilon Y$, then ϕ is *time-independent,* and we may as well write ϕ:$Y\to\mathbb{R}$; most useful priority rules are in fact of this type. Time-independent rules are especially attractive here because packets need only be ranked once (*e.g.* inserted into a linked-list data structure); this of course represents less overall computation, but also facilitates consideration of packets arriving relatively close to the decision epoch.

Given a static problem $\langle t,\nu\rangle$ we define the sequence σ *generated by* a time-independent priority function ϕ (or the corresponding rule) by

$$\sigma_j = \{i\epsilon\Omega: \phi[\nu(i)] > \phi[\nu(j)]\} \cup \{i\epsilon\Omega: \phi[\nu(i)] = \phi[\nu(j)]\ \&\ i<j\} \qquad (3.24)$$

for all $j\epsilon\Omega$; the time-dependent case is tedious to write down, but should be intuitively clear as well. Given a decision cost g, we will say that a time-independent priority function (or rule) ϕ^* is optimal (in the static sense) if it generates an optimal sequence for every $\langle t,\nu\rangle \epsilon\ \mathbb{R}\times\Upsilon$. For certain decision costs, optimal time-independent rules do exist (we are aware of no time-dependent rules which are optimal in this sense). Similarly, the (stationary) policy π generated by a priority function ϕ (or rule) may be defined by

$$\mu(t,v) \equiv \min\{i \in Q: \phi[t,v(i)] \geq \phi[t,v(j)] \text{ for all } j \in Q\} \tag{3.25}$$

for all $x \in X$, where the notation emphasizes the fact that decisions depend only on t and v; similarly, for a time-independent rule we write $\mu(v)$. The policy generated by an optimal priority rule may be optimal as well for certain statistics P.

To illustrate these ideas, priority functions corresponding to a number of well-known selection rules (usually associated with single-queue systems) are now presented. First-come-first-served may be expressed as $\phi(y) = -a$. In the case of linear decision costs $g(t,y) = m_y t$, the so-called "μC" rule is given by $\phi(y) = m_y/\tau$; it is optimal for the static problem, as we will demonstrate below, and the policy it generates is optimal for the dynamic problem (in the stable case) when the arrival process is Poisson and the classes and service times are i.i.d. and independent [Lippman] [see also Fife]. In the special case $g(t,y) = t$, the "shortest job first" rule mentioned in §2.3 is of course given by $\phi(y) = 1/\tau$. Now consider costs of the form $c(b,t) = \max\{0, t-T_b\}$; T_b has the interpretation of a due-date or grace period, after which a linear cost is incurred. Jackson's "due-date rule" is given by $\phi(y) = -(T_b - \eta)$, *i.e.* select the packet whose due-date is closest; for a single queue, the rule can be expressed as $\phi(y) = -(T_b + a)$, and minimizes the "maximum lateness" $\max_N\{c(b_n, s_n)\}$ in the static problem [Conway *et al*; §§3.3, 8.8]. Finally, Kleinrock's time-dependent rule [1976; §3.7] is the one used by Wong *et al*. to place the means of the various route delays (see §2.3.2 above); it is probably most meaningfully expressed here as $\phi(t,y) = m_r(t-a)$, where the m_r are suitable constants, chosen independently for every link.

It is easy to show that, in the case of linear decision costs $g(t,y) = m_y t$, the μC rule

$$\phi^*(y) = m_y/\tau \tag{3.26}$$

is optimal (in the static sense).[34] Let $\langle t, \nu \rangle$ be given, and let σ^* be the sequence generated by ϕ^*; without loss of generality, assume that $\Omega = \{1, 2, \ldots q\}$ and $\sigma^*(1) < \sigma^*(2) < \ldots < \sigma^*(q)$; then $\Gamma(\sigma^*, t, \nu) \leq \Gamma(\sigma, t, \nu)$ for every σ, as follows. If packet 1 is not first in σ, we can interchange it with the immediately preceeding packet j without increasing the total cost: the costs of the other packets clearly are unaffected by the interchange (because $\tau_1 + \tau_j = \tau_j + \tau_1$); the cost of packet 1 decreases by $m_1 \tau_j$ and the cost of packet 2 increases by $m_j \tau_1$, and $m_1/\tau_1 \geq m_j/\tau_j$ yields for the net cost increase $m_j \tau_1 - m_1 \tau_j \leq 0$. Iterating the argument, packet 1 can be "bubbled" to the first position in the sequence without increasing the cost, and packets $2, 3, \ldots q-1$ can similarly be bubbled to the 2nd, 3rd, ... (q-1)st positions without increasing the cost. In general, the optimality of any time-independent rule may in principle be established through such an interchange argument, because the rule will give an optimal sequence for the two-packet problem corresponding to any interchange pair.

It might seem, at first glance, that an optimal priority rule should always generate an optimal policy, but this is not so. The following example is a little involved, but illustrates some pertinent points. Consider a single queue with linear costs $c(b,t) = m_b t$, using the μC rule $\phi(y) = m_b/\tau$. Suppose that, for a given realization, packets 1, 2, and 3 have the following transmission times, delay costs, and priority assignments:

$\tau_1 = \tau$	$c(b_1,t) = t$	$\phi(y_1) = 1/\tau$
$\tau_2 = (1+2\epsilon)\tau$	$c(b_2,t) = (1+\epsilon)t$	$\phi(y_2) = (1+\epsilon)/(1+2\epsilon)\tau$
$\tau_3 = \tau$	$c(b_3,t) = 2t$	$\phi(y_3) = 2/\tau$

[34]Although the proof is simple, it is new to us. Proofs of optimality in the dynamic case proceed differently.

where $\epsilon << 1$. Packets 1 and 2 arrive simultaneously, and packet 3 arrives $(1+2\epsilon)\tau$ seconds later. Then the policy generated by the μC rule (which is indeed optimal for packets 1 and 2 only) results in the sequence 1-2-3, with total cost $(7+4\epsilon+2\epsilon^2)\tau$; on the other hand, the sequence 2-3-1 has total cost $(6+5\epsilon+2\epsilon^2)\tau$.

It is fairly clear that the scheduler is forced into making a poor decision because of the variation in transmission times (the example suggests as well the possible benefits of inserted idle time); indeed, if we reset $\tau_n = \tau$ for all n, the μC rule reduces to a "maximum slope" rule which gives the optimal sequence 2-1-3; similarly, if $\tau_n = (1+2\epsilon)\tau$ for all n, the rule correctly gives 2-3-1. It turns out that, in the case of constant transmission times, every optimal time-independent rule always (*i.e.* for every P) generates an optimal policy (in fact, it minimizes the cost for every realization); the proof is based *on interchanges of non-adjacent packets*, and is given in §A.3. Thus in the case of linear decision costs, the maximum slope rule generates an optimal policy for the G/D/1 link.

3.4 Selection Rules for Strictly Convex Costs

The μC rule may in some cases be a useful guideline even when the delay costs are not linear. Given a problem $\langle t, v \rangle$ in the single-queue case, if the marginal costs m(b,) can be reasonably approximated as constant over the corresponding intervals $[\eta, \eta + \Sigma_Q \tau_n]$, then the interchange argument of §3.3.2 has some validity, and the use of the priority rule

$$\phi(t,y) = m[b, \eta(t,y)]/\tau \tag{3.27}$$

---

[25]Moreover, there probably exist coupled constraints on the ranges of the marginal costs and transmission times under which an optimal sequence is guaranteed.

should result in good performance[25] (note that here $\eta = t-a$). We should also expect good performance in the dynamic problem [*cf.* Haji *et al.*] if the marginal costs and queue size are such that the linear approximation is valid most of the time. In the many-link case we have[26] $g'(t,y) = \partial g(t,y)/\partial t = \beta_y m(b, \beta_y t + K_y)$ from (3.17), and the corresponding rule is

$$\phi(t,y) = \beta_y m(b,\eta)/\tau, \tag{3.28}$$

with η as in (3.13); in the special case of linear costs, we get $\phi(y) = \beta m_b/\tau$, again a time-independent rule.

If we specialize to constant transmission times τ, we can find optimal time-independent rules in some other cases of interest. Let c_0:IR→IR be any CND function, with derivative m_0:IR→IR, and consider the case of identical costs $c(b,t) = c_0(t)$, $t \geq 0$, in a single-queue system. In this case the optimal rule is $\phi(t,y) = \eta$ (*cf.* §3.2.3); this is equivalent to FCFS

$$\phi(y) = -a \tag{3.29}$$

[*cf.* Henderson *et al*; Thm. 2] [*cf.* Haji *et al*], and thus generates an optimal policy for the G/D/1 queue. To show this, suppose that in some sequence we have ℓ arbitrary packets followed by packet j and then packet i, and that $\eta_i \geq \eta_j$. If we interchange i and j, the cost of packet i decreases by $c_0(\eta_i + \ell\tau + \tau) - c_0(\eta_i + \ell\tau)$ and that of packet j increases by $c_0(\eta_j + \ell\tau + \tau) - c_0(\eta_j + \ell\tau)$; the net increase is then

$$\int_{\eta_j+\ell\tau}^{\eta_j+(\ell+1)\tau} m_0(t)dt \quad - \quad \int_{\eta_i+\ell\tau}^{\eta_i+(\ell+1)\tau} m_0(t)dt \quad \leq \quad 0 \tag{3.30}$$

---

[26]there is of course an implicit time-dependence in (3.14) [and similarly for S(b,r)], and in practice β_y might depend on t as well; we assume here that these parameters do not change significantly over large numbers of transmission times.

(for every $\tau>0$), because m_0 is nonnegative and nondecreasing. In fact, it is not hard to show that if $|\tau_i-\tau_j| \leq |\eta_i-\eta_j|$ for all $i,j \in Q$, then the rule makes optimal choices in the case of variable transmission times as well.

Now in a somewhat more practical vein, we can derive different delay costs from c_0 by shifts of the form $c(b,t) = c_0(t+\Delta_b) - c_0(\Delta_b)$, $t\geq 0$, as illustrated in Fig. 6. In this case $m(b,t) = m_0(t+\Delta_b)$, and it is not hard to see (or prove) that in a single queue the optimal rule is simply $\phi(t,y) = \Delta_b + \eta$, or

$$\phi(y) = \Delta_b - a; \tag{3.31}$$

again the rule generates an optimal policy for the G/D/1 queue. Note that the due-date scenario may be expressed as the particular case $c_0(t) = \max\{0,t\}$, with $\Delta_b = -T_b$. But while the careful construction of a nominal cost c_0 might yield a useful collection of delay costs $c(b,\ )$, there is no guarantee of this; in particular the scaling requirement is not likely to be met.

In the many-link case, $c(\Delta_b)$ is time-independent, and we may assume $g(t,y) = c_0(\eta+\tau+\Delta_b)$. Then if we use the time-independent future estimate (3.15), it is again easy to see that the rule

$$\phi(y) = \Delta_b + h - a + f \tag{3.32}$$

is optimal, and generates an optimal policy for a G/D/1 link. However, with the more realistic time-dependent future estimate (3.16), we have $g(t,y) = c_0(\beta_y t + K_y + \Delta_b)$ from (3.17), and the decision costs no longer correspond to a simple family of shifted functions. In particular $g'(t,y) = \beta_y m_0(\beta_y t + K_y + \Delta_b)$; because the time-scaling of these marginal decision costs depends on y, we in

---

[17] We can improve the pessimistic estimate (3.15) without introducing time dependence, by replacing (t-a) in (3.16) with an estimate L(b,r) of the queueing delay on the given link; but there remains a tradeoff between the quality of the estimate and our ability to characterize good policies.

general cannot make the same kind of argument as in (3.30), and cannot find an optimal time-independent rule.[27]

In general, it is hard to say much about the nature of optimal rules when the transmission times are not constant. The due-date rule $\phi(t,y) = -(T_b - t)$ is one exception; and while the maximum lateness $\max_N\{c(b_n, s_n)\}$ is not an additive performance measure, it is appropriate here as the limiting case $\{\Sigma_N [c(b_n, s_n)]^p\}^{1/p}$ as $p \to \infty$. With a time-independent estimate f, the form of the decision costs does not change in the network case; thus the rule

$$\phi(y) = -[T_b - (h - a + f)], \tag{3.33}$$

a special case of (3.32), is optimal in the static sense for variable transmission times as well. As another case of interest, Schrage [1973] has derived a suboptimal time-dependent rule for variable service times to reduce the second moment of delay [*i.e.* $c(b,t) = t^2$]; the results of his simulation for an M/M/1 queue with utilization $\rho = 0.95$ showed substantial improvement over FCFS. In our notation the rule is

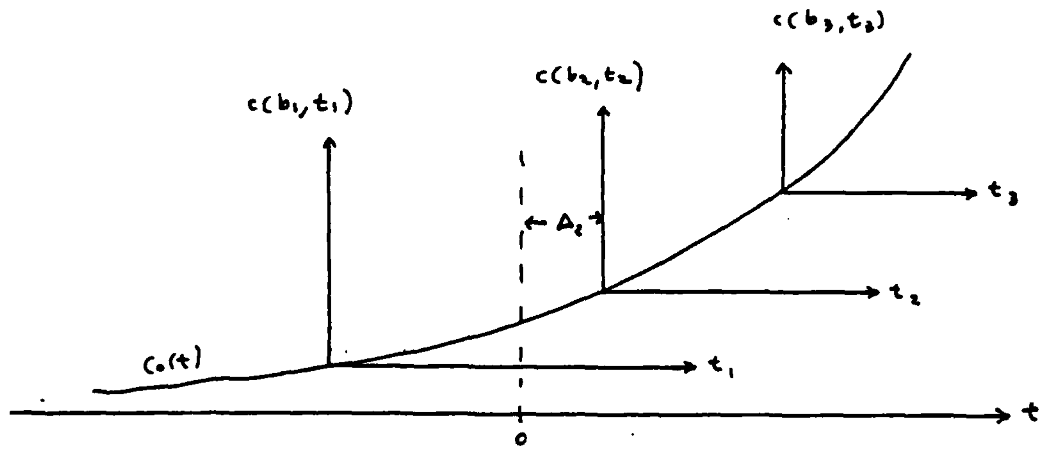

Figure 6 Shifted versions of a given nominal cost

$$\phi(t,y) = 2\eta E^2(\tau)/\tau - \tau^2, \tag{3.34}$$

where $E(\tau)$ is the expected transmission time overall, and is similarly applicable in the network case with a time-independent estimate f.

Unfortunately, we have not found reasonable policies for the important case in which competing cost functions have very different forms. Part of the problem is that the interchange argument is not reliable in general, as suggested earlier. Now consider the set Σ of all schedules σ for a given problem $<t,v>$. The success of the interchange argument for an optimal time-independent rule rests on the fact that every local minimum of $\Gamma(\ ,t,v):\Sigma \to \mathbb{R}$ is a global minimum, *i.e.* Γ is in some sense "convex" in Σ. However this property does not hold in general. Conway *et al.* give a counterexample in which the service times are variable; the example below shows that the property can fail for constant transmission times as well.

Suppose that packets 1, 2, and 3 have transmission times τ, and arrive simultaneously at a single queue system; let $c(b_1,t) = t$ and $c(b_2,t) = c(b_3,t) = a\max\{0, t-2\tau\}$, with $a >> 1$. Then the sequences 1-2-3 and 1-3-2 each have total cost $a\tau$, and are local minima; in either case, if packet 1 is interchanged with the succeeding packet, the cost increases to $(\epsilon + a)\tau$. Yet the global optima are the sequences 3-2-1 and 2-3-1 with cost 2ϵ. Thus a selection rule may be myopic as well, in the sense that it can "paint itself into a corner" by not looking at the entire cost function.

CHAPTER IV - SUMMARY AND CONCLUSION

We have addressed the issue of delay performance in packet networks by assigning delay costs for different classes of traffic, and adopted a limiting-average expected cost approach. In particular, we suggest that applications characterized by constraints on individual packet (or round-trip) response times lead to strictly convex costs, while applications for which job-related constraints dominate will have linear costs. We showed that the expected cost in equilibrium is an equivalent performance measure, and characterized the equilibrium delay performance of a typical network; this suggested considerable potential for reducing convex costs by packet scheduling, especially when link traffic is relatively heavy. In contrast, routing seemed to be relatively inflexible in this respect.

A distributed scheduling implementation was proposed, which incorporates knowledge of packet histories and estimates of packet futures; we know little about the tradeoff between resolution and communication overhead associated with time stamps for packet histories. Link scheduling was formulated as a dynamic programming problem, with decision costs representing estimates of the delay cost on system exit. We concentrated on simple static selection rules appropriate in a real-time environment, but showed that for some idealized cases such rules are optimal. In particular, we found that, in the case of constant transmission times, a time-independent priority rule which is optimal in the static sense always generates an optimal dynamic policy. The most useful case we found was that of shifted delay costs of the form $c(b,t) = c_0(t + \Delta_b)$, where c_0 is convex and nondecreasing; in this case the rule $\phi(y) = \Delta_b + h - a + f$ is optimal for a G/D/1 link, if the downstream estimate is time-independent. We did not find useful rules for the important case in

which the delay costs have radically different forms, but helped explain why this case is more difficult.

We believe that the basic idea potentially has practical merit, especially as processing costs come down in the future.

APPENDIX

A.1 Convergence of Average Cost to Equilibrium Cost

<u>Theorem:</u> Suppose a sequence $\{F_n\}$ of distribution functions converges weakly to a distribution function F, and let c: IR → IR be nonnegative, nondecreasing, and continuous. If

$$\lim_{A \to \infty} \int_{|t|>A} c(t)dF_n(t) = 0 \quad \text{uniformly in } n, \tag{A.1}$$

then

$$\lim_{N \uparrow} (1/N) \sum_{n=1}^{N} \int c(t)dF_n(t) = \int c(t)dF(t) \tag{A.2}$$

<u>Remarks:</u> Note that if F_n is nonnegative [*i.e.* $F_n(t) \equiv 0$ for $t<0$] for all n, then so is F, and we are only concerned about the behavior of c on $[0,\infty)$. Thus if we substitute $F_j(;b)$ for F_n, $F(;b)$ for F, and $c(b,)$ for c, (A.2) is equivalent to (2.14).

In fact it is easy to show that if $\int c(t)|t|^p dF_n(t)$ is uniformly bounded for some p>0, then (A.1) holds [use the approach on p. 186 of Loeve]; thus uniform boundedness of the expected costs $E\{c[b,s(n_j)]\}$ is "almost" sufficient for (2.14).

<u>Proof:</u> Denote by C_n and C the expectations with respect to F_n and F in (A.2); we first show[28] that $C_n \to C$. Define for every $A \in$ IR the function c_A by

[28]This is a known result [Loeve; §11.4]; the proof here is based on a moment convergence proof from Chung [Thm. 4.5.2], and is more direct.

$$c_A(t) \equiv \begin{cases} c(t) & \text{if } t \le A \\ c(A) & \text{if } t > A \end{cases} \tag{A.3}$$

These functions are all continuous and bounded on IR, so we have for all A

$$\lim_{n\uparrow} \int c_A(t)dF_n(t) = \int c_A(t)dF(t) \tag{A.4}$$

directly from the definition of weak convergence [Ash; §§4.5, 8.1]. Also,

$$\int [c(t)-c_A(t)]\, dF_n(t) \;=\; \int_{t>A} [c(t)-c_A(t)]\, dF_n(t)$$

$$\le \int_{t>A} c(t)dF_n(t) \;\le\; \int_{|t|>A} c(t)dF_n(t), \tag{A.5}$$

so by (A.1), $\int(c-c_A)dF_n$ vanishes uniformly, *i.e.*

$$\lim_{A\uparrow} \int c_A(t)dF_n(t) = C_n \tag{A.6}$$

uniformly in n.

It then follows from (A.4) and (A.6), and properties of uniform convergence [*cf.* Rudin; p. 149], that $\int c_A dF$ converges and

$$\lim_{A\uparrow} \int c_A(t)dF(t) \;=\; \lim_{n\uparrow} C_n, \tag{A.7}$$

i.e. the iterated limits exist and are equal. But $\int c_A dF \to C$ by (essentially) monotone convergence [Ash; p. 44], so $C_n \to C$.

Thus for every $\epsilon>0$, there exists an M such that $|C_n-C| < \epsilon/2$ for $n \ge M$. Take

$$L \equiv \left\lceil \max\{(2/\epsilon) \sum_{n=1}^{M-1} (C_n-C),\ M\} \right\rceil \tag{A.8}$$

where $\lceil x \rceil$ is the smallest integer not less than x; then for N≥L, we have

$$\left|(1/N)\sum_{n=1}^{N}C_n - C\right|$$

$$= \left|(1/N)\sum_{n=1}^{N}(C_n - C)\right|$$

$$= \left|(1/N)\sum_{n=1}^{M-1}(C_n - C) + (1/N)\sum_{n=M}^{N}(C_n - C)\right|$$

$$\leq \left|(1/L)\sum_{n=1}^{M-1}(C_n - C)\right| + \left|(N-M+1)^{-1}\sum_{n=M}^{N}(C_n - C)\right|$$

$$\leq (\epsilon/2) + (\epsilon/2) = \epsilon, \tag{A.9}$$

i.e. $(1/N)\Sigma_N C_n \to C$, which is equivalent to (A.2).

A.2 Alternate Route Delays in an Optimal Flow Assignment

We consider here, in a network setting, the traffic leaving a given node, intended for a given destination; two alternate paths p_1 and p_2, of N_1 and N_2 hops, are available. We model the equilibrium behavior with a function ν, such that $\nu(f)$ gives the expected number of packets (in queue and transmitter) on a link with flow f, independently at all links in the network. The expected number of packets in the system is then the sum $L = \Sigma_E \nu(f_e)$ over all links $e \in E$, and by Little's formula [*cf.* Stidham, 1972; pp. 1122-23] we have $L = \lambda S$, where λ is the total arrival rate. Thus a minimum-delay flow assignment is one which minimizes $\Sigma_E \nu(f_e)$; here we examine some of the implications of such an assignment.

If we assume further that ν is convex (as it is for M/G/1), we can characterize the optimal flow assignment as follows [Bertsekas, 1981; eqn. (4.4) and below]: if $\nu'(f) = \partial\nu(f)/\partial f$, and we define the *first derivative length* (FDL) of a

path p by $\Sigma_p \nu'(f_e)$, then *path flow is positive only on paths with minimum FDL.* Thus if we have positive flow on both p_1 and p_2, optimality requires that

$$d_1 = \sum_{p1} \nu'(f_e) = \sum_{p2} \nu'(f_e) = d_2. \qquad \text{(A.10)}$$

Intuitively, if p_1 had a smaller FDL, then some flow could be shifted from p_2 to p_1, and by convexity the sum $\Sigma_{p1}\nu(f_e) + \Sigma_{p2}\nu(f_e)$ would decrease, equivalent to a decrease in overall delay.

If the derivative ν' is convex as well (also true for M/G/1), we expect that, for a given first derivative length, variability among link flows along a path implies a smaller average link flow (over the path) than does a relatively constant link flow profile. But because ν itself is convex, variability among link flows also requires less average link flow for a given number of packets on the path, and when $N_1 = N_2$ these factors tend to cancel. As an example, a string of four M/M/1 links, with expected transmission times $E(\tau)$, capacities θ, and flows 0.4θ, 0.5θ, 0.6θ, and 0.7θ, has a path delay of $S_1 \simeq 9.5E(\tau)$; the same system has a path delay of $S_2 \simeq 9.8E(\tau)$ if all link flows are equal and adjusted such that the FDLs are equal, $d_2 = d_1$.

If $N_1 \neq N_2$, the path delays are not as constrained. In the case of M/M/1 links with constant path flows f_1 and f_2, optimality requires

$$N_1\theta/(\theta-f_1)^2 = N_2\theta/(\theta-f_2)^2, \text{ or} \qquad \text{(A.11)}$$

$$\sqrt{N_1}/(\theta-f_1) = \sqrt{N_2}/(\theta-f_2), \text{ which gives}$$

$$N_1E(\ell)\sqrt{N_2}/(\theta-f_1) = N_2E(\ell)\sqrt{N_1}/(\theta-f_2), \text{ or}$$

$$S_1\sqrt{N_2} = S_2\sqrt{N_1}, \qquad \text{(A.12)}$$

where $E(\ell)$ is the expected packet length. Thus the greater hop-length path has a larger path delay, but the relative difference is still not so large for N_2

close to N_1. The previous paragraph suggests that non-constant path flows should not significantly alter this result.

A.3 Optimal Priority-based Policies for G/D/1 Links

To formalize the G/D/1 case, let $\Upsilon_c \equiv \{\nu \epsilon \Upsilon: \tau(i) = \tau(j) \text{ for all } i,j \epsilon \Omega\}$ be the set of all queue lists with constant transmission times. Then given g, a priority rule ϕ^* is optimal in the case of constant transmission times if it generates an optimal sequence for every $<t,\nu> \epsilon\, \mathbb{R} \times \Upsilon_c$ (it is intuitively compelling that if a rule is optimal for any given constant τ, then it is optimal for every such constant; indeed, we show in §3.4 the optimality of some rules of this type, and the proofs are always independent of τ). Now if $\Omega_c \equiv \{\omega \epsilon \Omega: \tau_i(\omega) = \tau_j(\omega) \text{ for all } i,j \epsilon \mathbb{N}^+\}$ is the set of all realizations with constant transmission times, we define $J_N(\pi | \Omega_c)$ and $J(\pi | \Omega_c)$ by conditioning the expectations[29] in (3.10) and (3.20) on Ω_c. Then we have the following:

<u>Theorem:</u> Given a decision cost g, suppose there exists a time-independent priority function ϕ^* which is optimal in the case of constant transmission times. Then for every P, in the problem <P,g> the stationary policy $\pi^* = <\mu^*,\mu^*,\ldots>$ generated by ϕ^* satisfies

$$J_N(\pi^* | \Omega_c) \le J_N(\pi | \Omega_c) \tag{A.13}$$

for every $\pi \epsilon \Pi$. Furthermore, if $J(\pi^* | \Omega_c)$ exists and is finite, then

$$J(\pi^* | \Omega_c) \le \liminf_{N \uparrow} (1/N) \{J_N(\pi | \Omega_c) + E[\Gamma^*_N(\pi) | \Omega_c]\} \tag{A.14}$$

for every $\pi \epsilon \Pi$, where $\Gamma^*_N(\pi)$ abbreviates $\Gamma^*[t_N(\pi), \nu_N(\pi)]$.

---

[29] If necessary, we can formalize the expectations by simply considering a new probability measure P_c associated with Ω_c.

Proof: Consider the problem of minimizing

$$j_L(\omega,\pi) \equiv \sum_{k=0}^{L-1} g\{t_k(\omega,\pi), y[u_k(\omega,\pi)]\} + \Gamma^*{}_L(\omega,\pi) \tag{A.15}$$

over $\pi\epsilon\Pi$, where $\Gamma^*{}_L(\omega,\pi)$ abbreviates $\Gamma^*[t_L(\omega,\pi),\nu_L(\omega,\pi)]$; j_L is the incurred cost plus liability after L decisions, for a given realization (assuming no further arrivals). We show below that, in fact,

$$j_L(\omega,\pi^*) \leq j_L(\omega,\pi) \tag{A.16}$$

for every $\omega\epsilon\Omega_c$, $\pi\epsilon\Pi$, L. Now in the finite case, Q_N is empty for every ω, π, so $\Gamma^*(t_N,\nu_N) = 0$; then with L = N in (A.16), taking expectations conditioned on Ω_c preserves the inequality [*cf.* Ash; p. 41], giving (A.13). In the infinite case, take expectations in (A.16) conditioned on[39] Ω_c, divide by L, and take lim inf as $L\to\infty$, each of which preserves the inequality; this gives (A.14) when the LHS converges.

Now let $\omega\epsilon\Omega_c$ and L be given, and suppose that $\tau_n(\omega) = \tau$ for all n: to get (A.16), we show that for every policy $\pi^0\epsilon\Pi$, there exists a sequence of policies π^0, π^1, ... π^{L-1}, π^*, in which j_L is non-increasing, *i.e.* $j_L(\omega,\pi^0) \geq j_L(\omega,\pi^1) \geq ... \geq j_L(\omega,\pi^*)$. To see how j_L changes from one policy to the next, it is helpful to visualize the packet selections u_0, u_1, ... u_{L-1} (dictated by a given policy) laid out along the time axis at epochs t_0, t_1, ... t_{L-1} (note that for constant transmission times, the t_k are independent of the policy, for a given realization); follow these by the packets in Q_L laid out according to the optimal static sequence σ^* generated by ϕ^*, at epochs t_L, $t_L+\tau$, ... $t_L+(q-1)\tau$ (where $q = |Q_L|$ is also independent of the policy). The basic idea is that, if

---

[39]Note that we could just as easily have conditioned on a given initial state here to get the more standard dynamic programming statement.

the packet u_j selected at t_j under the policy π^j is not the one dictated by μ^*, *i.e.* if $u_j(\pi^j) \neq \mu^*[\nu_j(\pi^j)]$, then we find a new policy π^{j+1}, under which the positions of packets $u_j(\pi^j)$ and $\mu^*[\nu_j(\pi^j)]$ in the ordering are interchanged; the effect of the transformations is that policy π^j has the form $<\mu^*, \ldots \mu^*, \mu^j{}_j, \mu^j{}_{j+1}, \ldots>$ $(j = 0, 1, \ldots L-1)$. Note that every such interchange must occur within some busy period (because μ^* is admissible); the lemma below shows that no such interchange can increase the value of j_L (the lemma is proved after the theorem).

<u>Lemma:</u> Given a decision cost g, suppose there exists a time-independent priority rule ϕ^* which is optimal in the case of constant transmission times; let μ^* be the map which defines the policy generated by ϕ^*. Then if $\nu \epsilon Y_C$ is a queue list with transmission times τ, $\mu^*(\nu) = i \epsilon Q$ implies

$$g[t, \nu(i)] + g[t+(m+1)\tau, \nu(j)] \leq g[t, \nu(j)] + g[t+(m+1)\tau, \nu(i)] \tag{A.17}$$

for every $i, j \epsilon Q$, $t \epsilon IR$, $m = 0, 1, \ldots$. Thus if packets i and j are separated by m packets in an ordering, the combined cost of both packets is no greater with packet i in the earlier position.

Formally, we get π^{j+1} from π^j as follows, for $j+1 = 1, \ldots L-1$:

<u>Case A:</u> $\mu^*[\nu_j(\pi^j)] = u_j(\pi^j)$; *i.e.* policy π^j makes the choice dictated by μ^* at epoch t_j, so simply pick $\pi^{j+1} = \pi^j$, and j_L does not change.

<u>Case B:</u> $\mu^*[\nu_j(\pi^j)] = u_p(\pi^j)$ for some $p \epsilon \{j+1, \ldots L-1\}$; *i.e.* the packet which μ^* would have chosen at t_j is instead selected at some later epoch $t_p < t_L$ by π^j. Then pick any π^{j+1} such that

$$\mu^{j+1}{}_j[x_j(\pi^{j+1})] = \mu^*[v_j(\pi^j)],$$

$$\mu^{j+1}{}_p[x_p(\pi^{j+1})] = \mu^j{}_j[x_j(\pi^j)],$$

$$\mu^{j+1}{}_k[x_k(\pi^{j+1})] = \mu^j{}_k[x_k(\pi^j)],\ k \neq j,p \tag{A.18}$$

(thus π^{j+1} is not uniquely specified here). Some thought shows that these are all admissible choices if π^j is admissible. Clearly the packets in Q_L are not affected by this interchange, so $\Gamma^*{}_L(\pi^{j+1}) = \Gamma^*{}_L(\pi^j)$; similarly, $g_k(\pi^{j+1}) = g_k(\pi^j)$ for $k \leq L-1$, $k \neq j,p$. Only the costs of packets $u_j(\pi^j)$ and $u_p(\pi^j)$ are affected, and by the lemma (with $m+1 = p-j$) we have

$$g\{t_j, y[u_j(\pi^{j+1})]\} + g\{t_p, y[u_p(\pi^{j+1})]\}$$
$$\leq g\{t_j, y[u_j(\pi^j)]\} + g\{t_p, y[u_p(\pi^j)]\}. \tag{A.19}$$

Thus $j_L(\pi^{j+1}) \leq j_L(\pi^j)$.

<u>Case C:</u> $\mu^*[v_j(\pi^j)] \epsilon Q_L(\pi^j)$; *i.e.* the packet that would have been selected at t_j by μ^* is not chosen by π^j in the first L decisions. In this case pick any π^{j+1} which satifies

$$\mu^{j+1}{}_j[x_j(\pi^{j+1})] = \mu^*[v_j(\pi^j)],$$

$$\mu^{j+1}{}_k[x_k(\pi^{j+1})] = \mu^j{}_k[x_k(\pi^j)],\ k \neq j. \tag{A.20}$$

Now if σ^* is the (optimal) sequence for $v_L(\pi^j)$ generated by ϕ^*, let σ' be the sequence for $v_L(\pi^{j+1})$ obtained by substituting $u_j(\pi^j)$ for $\mu^*[v_j(\pi^j)]$ in σ^*; then assuming ℓ packets preceed $\mu^*[v_j(\pi^j)]$ *in* σ^* *[or* $u_j(\pi^j)$ *in* σ'*], the lemma* gives (with $m+1 = L-1+\ell$)

$$g\{t_j, y[u_j(\pi^{j+1})]\} + g\{t_L+\ell\tau, y[u_j(\pi^j)]\}$$
$$\leq g\{t_j, y[u_j(\pi^j)]\} + g\{t_L+\ell\tau, y[u_j(\pi^{j+1})]\}. \tag{A.21}$$

Since $g_k(\pi^{j+1}) = g_k(\pi^j)$ for $k \le L-1$, $k \ne j$, and the costs $g(t_L + |\sigma^*{}_n|\tau, y_n)$ for all other packets $n \in Q_L$, $n \ne \mu^*[\nu_j(\pi^j)]$, are unaffected by the interchange, we have

$$\sum_{k=0}^{L-1} g_k(\pi^{j+1}) + \Gamma^*{}_L(\pi^{j+1}) \le \sum_{k=0}^{L-1} g_k(\pi^{j+1}) + \Gamma[\sigma', t_L, \nu_L(\pi^{j+1})]$$

$$\le \sum_{k=0}^{L-1} g_k(\pi^j) + \Gamma[\sigma^*, t_L, \nu_L(\pi^j)] \qquad \text{(A.22)}$$

(where the first inequality follows from the definition of Γ^*), or in other words $j_L(\pi^{j+1}) \le j_L(\pi^j)$.

We get π^* from π^{L-1} in essentially the same way, in each case, except that for $k = L, L+1, \ldots$ we must of course set $\mu^*{}_k = \mu^*$, which clearly has no effect on j_L.

<u>Proof of Lemma:</u> The lemma is a statement about only packets i and j; to prove it we define a new queue list $\nu' \epsilon \Upsilon_c$ consisting of packets i and j, and m other packets whose characteristics are the same as those of j. For concreteness, let $n_0 \equiv \max\{i,j\} + 1$ and $Q' \equiv \{i, j, n_0, \ldots n_0+m-1\}$; then $\nu'(i) \equiv \nu(i)$, and $\nu'(n) \equiv \nu(j)$ for $n = j, n_0, \ldots n_0+m-1$.

Now consider the sequence defined by

$$\sigma(j) < \sigma(n_0) < \ldots < \sigma(n_0+m-1) < \sigma(i). \qquad \text{(A.23)}$$

The first two packets in the sequence (j and n_0+m-1) are indistinguishable, and interchanging them clearly cannot affect the total cost $\Gamma(\sigma,t,\nu')$; similarly we can bubble packet j to the next-to-last position for the same total cost. Now we can interchange packets i and j without increasing the total cost: simply note that i precedes j in the optimal sequence for the corresponding two-packet problem. Iterating the same argument, we can bubble packet i to the first position without increasing the total cost. Finally, packets

n_0, ... n_0+m-1 are now back in their original positions, so their combined cost is the same, hence the combined cost of packets i and j cannot have increased.

REFERENCES

Reference works are ordered by keyword (usually the author's last name) and year. Citations in the text include the keyword, year if necessary, and and location wihin the work if appropriate. Starred items are not cited in the text.

R.B. Ash, *Real Analysis and Probability,* Academic Press, New York, 1972.

R.B. Ash and M.F. Gardner, *Topics in Stochastic Processes,* Academic Press, New York, 1975.

Bell Telephone Laboratories Technical Staff, *Transmission Systems for Communication,* rev. 4th ed., Western Electric, Winston-Salem NC, 1971.

D.P. Bertsekas, *Dynamic Programming and Stochastic Control,* Academic Press, New York, 1976.

____________, "Notes on Optimal Routing and Flow Control for Communication Networks," MIT Lab for Information and Decision Systems LIDS-R-1169, Dec. 1981.

K.L. Chung, *A Course in Probability Theory,* Academic Press, New York, 1974.

R.W. Conway, W.L. Maxwell, and L.W. Miller, *Theory of Scheduling,* Addison-Wesley, Reading MA, 1967.

S.E. Elmaghraby, "The One Machine Sequencing Problem with Delay Costs," *J. Industrial Engineering* v. 19, Feb. 1969, pp. 105-108.

D.W. Fife, "Scheduling with Random Arrivals and Linear Loss Functions," *Management Science* v. 11 #3, Jan. 1965, pp. 429-37.

R.G. Gallager, "A Minimum Delay Routing Algorithm using Distributed Computation," *IEEE Trans. Communications* v. COM-25 #1, Jan. 1977, pp. 73-85.

M. Gerla and L. Kleinrock, "On the Topological Design of Distributed Computer Networks," *IEEE Trans. Communications* v. COM-25 #1, Jan. 1977, pp. 48-60.

____________ and ____________, "Flow Control: A Comparative Survey," *IEEE Trans. Communications* v. COM-28 #4, Apr. 1980, pp. 553-74.

R. Haji and G.F. Newell, "Optimal Strategies for Priority Queues with Nonlinear Costs of Delay," *SIAM J. Applied Mathematics* v. 20 #2, Mar. 1971, pp. 224-40.

F.E. Heart, R.E. Kahn, *et al.*, "The Interface Message Processor for the ARPA Computer Network," *AFIPS 1970 Spring Joint Computer Conf.* v. 36, pp. 551-67.

M. Held and R.M. Karp, "A Dynamic Programming Approach to Sequencing Problems," *SIAM J. Applied Mathematics* v. 10 #1, Mar. 1962, p. 196.

P.B. Henderson and K. Steiglitz, "A Problem in Single-Machine Sequencing with Nonlinear Delay Costs," *Management Science* v. 22 #12, Aug. 1976, pp. 1342-50.

D.P. Heyman and M.J. Sobel, *Stochastic Models in Operations Research, Volume I: Stochastic Processes and Operating Characteristics,* McGraw-Hill, New York, 1982.

____________ and ____________, *Stochastic Models in Operations Research, Volume II: Stochastic Optimization,* McGraw-Hill, New York, 1984.

O.C. Ibe, "Flow Control and Routing in an Integrated Voice and Data Communication Network," MIT Lab for Information and Decision Systems LIDS-TH-1115 (Ph. D. thesis), Aug. 1981.

*H. Kesten and J. Th. Runnenburg, "Priority in Waiting Line Problems" (Parts I and II), *Proc. (Koninklijke Nederlandse) Akademie van Wetenschappen* v. 60 (series A), North-Holland, Amsterdam, 1957, pp. 312-24, 325-35.

L. Kleinrock, *Queueing Systems, Volume I: Theory,* Wiley, New York, 1975.

____________, *Queueing Systems, Volume II: Computer Applications,* Wiley, New York, 1976.

E.L. Lawler, "On Scheduling Problems with Deferral Costs," *Management Science* v. 11 #2, Nov. 1964, pp. 280-88.

____________, *Combinatorial Optimization: Networks and Matroids,* Holt, Rinehart and Winston, New York, 1976.

S.A. Lippman, "On Dynamic Programming with Unbounded Rewards," *Management Science* v. 21 #11, July 1975, pp. 1225-33.

M. Loeve, *Probability Theory I,* 4th ed., Springer-Verlag, New York, 1977.

L.G. Roberts and B.D. Wessler, "Computer Network Development to Achieve Resource Sharing," *AFIPS 1970 Spring Joint Computer Conf.* v. 36, pp. 543-49.

S.M. Ross, "Arbitrary State Markovian Decision Processes," *The Annals of Mathematical Statistics* v. 39 #6 (1968), pp. 2118-22.

W. Rudin, *Principles of Mathematical Analysis, 3rd ed.,* McGraw-Hill, New York, 1976.

A. Schild and I.J. Friedman, "Sheduling Tasks with Deadlines and Non-Linear Loss Functions," *Management Science* v. 9 (1962), pp. 73-81.

L. Schrage, "A Proof of the Optimality of the Shortest Remaining Processing Time Discipline," *Operations Research* v. 16 (1968), pp. 687-90.

____________, "Optimal Scheduling Rules for Quadratic Waiting Costs," *Seventh Annual Princeton Conf. on Information Sciences and Systems,* 1973, pp. 424-27.

S. Stidham, Jr., "L = λW: A Discounted Analogue and a New Proof," *Operations Research* v. 20 (1972), pp. 1115-26.

S. Stidham, Jr., and N.U. Prabhu, "Optimal Control of Queueing Systems," *Mathematical Methods in Queueing Theory,* ed. A.B. Clarke, Springer-Verlag, New York, 1974.

P. Suppes, *Axiomatic Set Theory,* Dover, New York, 1972.

*L. Takacs, "Priority Queues," *Operations Research* v. 12 #1, Jan/Feb. 1964, pp. 63-74.

A.S. Tanenbaum, *Computer Networks,* Prentice-Hall, Englewood Cliffs NJ, 1981.

J. Wong, J. Sauve, and J. Field, "A Study of Fairness in Packet-Switching Networks," *IEEE Trans. Communications,* v. COM-30 #2, Feb. 1982, pp. 346-53.

Distribution List

Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12 Copies
Assistant Chief for Technology Office of Naval Research, Code 200 Arlington, Virginia 22217	1 Copy
Office of Naval Research Information Systems Program Code 437 Arlington, Virginia 22217	2 Copies
Office of Naval Research Branch Office, Boston 495 Summer Street Boston, Massachusetts 02210	1 Copy
Office of Naval Research Branch Office, Chicago 536 South Clark Street Chicago, Illinois 60605	1 Copy
Office of Naval Research Branch Office, Pasadena 1030 East Greet Street Pasadena, California 91106	1 Copy
Naval Research Laboratory Technical Information Division, Code 2627 Washington, D.C. 20375	6 Copies
Dr. A. L. Slafkosky Scientific Advisor Commandant of the Marine Corps (Code RD-1) Washington, D.C. 20380	1 Copy

Office of Naval Research
Code 455
Arlington, Virginia 22217

1 Copy

Office of Naval Research
Code 458
Arlington, Virginia 22217

1 Copy

Naval Electronics Laboratory Center
Advanced Software Technology Division
Code 5200
San Diego, California 92152

1 Copy

Mr. E. H. Gleissner
Naval Ship Research & Development Center
Computation and Mathematics Department
Bethesda, Maryland 20084

1 Copy

Captain Grace M. Hopper
Naval Data Automation Command
Code OOH
Washington Navy Yard
Washington, DC 20374

1 Copy

Advanced Research Projects Agency
Information Processing Techniques
1400 Wilson Boulevard
Arlington, Virginia 22209

1 Copy

Dr. Stuart L. Brodsky
Office of Naval Research
Code 432
Arlington, Virginia 22217

1 Copy

Prof. Fouad A. Tobagi
Computer Systems Laboratory
Stanford Electronics Laboratories
Department of Electrical Engineering
Stanford University
Stanford, CA 94305